



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1994-03

Lessons learned from the 14-year systems development of the Marine Corps' Standard Accounting, Budgeting and Reporting System (SABRS)

Tavares, Jeffrey Louis

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/42957>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

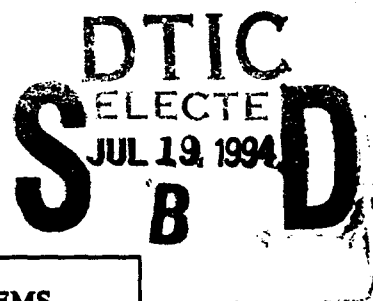
AD-A281 641



NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS



**LESSONS LEARNED FROM THE 14-YEAR SYSTEMS
DEVELOPMENT OF THE MARINE CORPS' STANDARD
ACCOUNTING, BUDGETING AND REPORTING SYSTEM
(SABRS)**

by

Jeffrey Louis Tavares

March 1994

Advisor:

Co-Advisor:

James C. Emery

Nancy C. Roberts

Approved for public release; distribution is unlimited.

94-22479



DTIC QUALITY INSPECTED 8

94 7 18 029

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE *March 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE* LESSONS LEARNED FROM THE 14-YEAR SYSTEMS DEVELOPMENT OF THE MARINE CORPS' STANDARD ACCOUNTING, BUDGETING AND REPORTING SYSTEM (SABRS)		5. FUNDING NUMBERS	
6. AUTHOR(S) *Jeffrey Louis Tavares			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE *A	
13. ABSTRACT (maximum 200 words) * In August of 1978 the Marine Corps initiated the development of a consolidated financial management system. On October 1, 1992, after 14-years of systems development effort, the Standard Accounting, Budgeting and Reporting System (SABRS) was finally implemented throughout the Marine Corps. This thesis chronicles the 14-year SABRS systems development effort using the historical case study research method. Data is presented from both archival sources and personal interviews. The SABRS project reveals some important general lessons about the systems development process that will prove useful to future project managers tasked with developing large-scale administrative information systems. These lessons learned include, but are not limited to, (1) the importance of top management support, (2) the role of the project manager as leader, rather than technical expert, (3) the use of adaptive prototyping, (4) the importance of fitting the right people to the right task, and (5) the ability of management to alter its commitment to a failed course of action.			
14. SUBJECT TERMS *Systems Development, Project Management, SABRS, Adaptive Prototyping, Structured Methodologies, Lessons Learned.		15. NUMBER OF PAGES * 109	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. Z39-18

Approved for public release; distribution is unlimited.

**LESSONS LEARNED FROM THE 14-YEAR SYSTEMS DEVELOPMENT OF THE MARINE
CORPS' STANDARD ACCOUNTING, BUDGETING AND REPORTING SYSTEM (SABRS)**

by

Jeffrey Louis Tavares
Captain, United States Marine Corps
B.S., United States Naval Academy, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

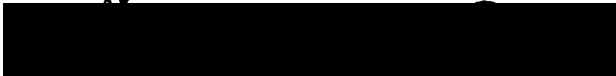
NAVAL POSTGRADUATE SCHOOL

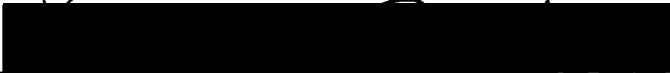
March 1994

Author:


Jeffrey Louis Tavares

Approved by:


Prof. James C. Emery, Advisor


Assoc. Prof. Nancy C. Roberts, Co-Advisor


Prof. David R. Whipple, Jr., Chairman,
Department of Systems Management

ABSTRACT

In August of 1978 the Marine Corps initiated the development of a consolidated financial management system. On October 1, 1992, after 14-years of systems development effort, the Standard Accounting, Budgeting and Reporting System (SABRS) was finally implemented throughout the Marine Corps. This thesis chronicles the 14-year SABRS systems development effort using the historical case study research method. Data is presented from both archival sources and personal interviews.

The SABRS project reveals some important general lessons about the systems development process that will prove useful to future project managers tasked with developing large-scale administrative information systems. These lessons learned include, but are not limited to, (1) the importance of top management support, (2) the role of the project manager as leader, rather than technical expert, (3) the use of adaptive prototyping, (4) the importance of fitting the right people to the right task, and (5) the ability of management to alter its commitment to a failed course of action.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. OBJECTIVE AND FOCUSING RESEARCH QUESTION	2
C. SCOPE, LIMITATIONS AND ASSUMPTIONS	3
1. Scope	3
2. Limitations	3
3. Assumptions	3
D. THESIS ORGANIZATION	4
II. THE SYSTEMS DEVELOPMENT PROCESS	6
A. INTRODUCTION	6
B. THE CLASSIC "WATERFALL" MODEL	6
1. Overview	6
2. Goals	7
a. Introduce Discipline	8
b. Improve Reliability and Reduce Errors	9
c. Use Resources More Efficiently	9
3. Problems	10
a. Too Much Documentation	10

b.	Requirements Are Not Stated Correctly	11
c.	Too Many Methodologies	11
C.	PROTOTYPING	12
1.	Overview	12
2.	Disadvantages	13
3.	Prototyping and Fourth-Generation Languages	14
D.	THREE GENERIC PHASES OF SYSTEMS DEVELOPMENT	15
E.	FACTORS CAUSING DELAYS, COST OVERRUNS, AND UNFULFILLED REQUIREMENTS	16
1.	General	16
2.	Shortcuts	16
3.	Analysis Paralysis	17
4.	Users Are Not Involved	17
5.	Unreasonable Demands	18
6.	System Complexity	18
7.	Inexperienced Technical and Managerial Personnel ..	19
F.	SUMMARY	20
III.	RESEARCH METHODOLOGY	21
A.	INTRODUCTION	21
B.	ARCHIVAL DATA COLLECTION	21
C.	PERSONAL INTERVIEW DATA COLLECTION	23
1.	Selection	23

2. Background	23
3. Interview Outline	25
D. DATA ANALYSIS	27
IV. GENESIS OF SABRS	28
A. PURPOSE	28
B. BACKGROUND	28
1. Problems With Existing Systems	28
2. Formulation of the SABRS Concept	31
C. SABRS OBJECTIVES	32
D. CONCEPT OVERVIEW	34
E. SUMMARY	36
V. SABRS DEVELOPMENT CHRONOLOGY	37
A. INTRODUCTION	37
B. SOURCE DOCUMENT MILESTONES AND ORGANIZATION	37
1. Concept Statement Milestones	37
2. Requirements Statement	38
3. Feasibility Study	39
4. Top Management Roles and Responsibilities	40
5. Automated Data System Development Plan (ADS DP) .	41
6. Analysis and Design Action Plan	42
C. THREE ERAS OF SABRS DEVELOPMENT	43

1.	The "Floundering" Era: 1978-1983	44
2.	The Era of Redirection and Progress: 1983-1987	44
3.	The Testing and Implementation Era: 1988-1992	45
D.	SUMMARY	46
VI.	SABRS INTERVIEW DATA	47
A.	PURPOSE	47
B.	REFERENCES	47
C.	ORGANIZATIONAL ENVIRONMENT	47
1.	Top Management Support	47
a.	Steering Committee	47
b.	SABRS Functional Manager	48
c.	C4 Project Officer	49
d.	Program Management	50
2.	User/Systems Development Team Relationship	52
D.	PROJECT TEAM	54
1.	Functional Area Qualifications	54
a.	Restructuring of the Financial Management Officer Community	54
b.	Unwillingness to Give Up Technical Experts	55
2.	Internal Organization	56
3.	External Organization	58
E.	DEVELOPMENT ENVIRONMENT	60

1.	Methodology Followed	60
a.	A Structured Methodology Was Not Followed During the "Floundering" Era	60
b.	Methodology Enforcement During the Era of Redirection and Progress	62
c.	Prototyping	63
2.	Tools	64
F.	APPLICATION CHARACTERISTICS	66
1.	Interaction With Other Systems	66
2.	Degree of Definition	67
3.	Technological Complexity	67
G.	POSTSCRIPT	68
H.	SUMMARY	69
VII.	OBSERVATIONS AND LESSONS LEARNED	70
A.	INTRODUCTION	70
B.	LESSONS LEARNED	70
1.	Top Management Support is Crucial to the Systems Development Process	70
2.	The Program Manager's Leadership Abilities Matter More Than His Technical Expertise	72
3.	A Technical Leader Must Emerge From Within the Funding Organization	74
4.	Structured Methodologies Only Help Organize the Process	75
5.	Adaptive Prototyping Can Save a Project	76

6. Get the Right People Involved	78
7. Do Not Become Dependent on an Uncertain Resource	80
8. Do Not Be Afraid to Start Over	82
VIII. SUMMARY AND RECOMMENDATIONS FOR FURTHER STUDY	85
A. SUMMARY	85
B. RECOMMENDATIONS FOR FURTHER STUDY	88
APPENDIX A - Factors That Influence Systems Delivery Rate	90
APPENDIX B - Focusing Interview Questions	93
LIST OF REFERENCES	95
INITIAL DISTRIBUTION LIST	98

I. INTRODUCTION

A. BACKGROUND

In August of 1978 the Marine Corps approved a plan by its Fiscal Division to develop a Standard Accounting, Budgeting and Reporting System (SABRS). Designed to replace a number of aging, "stovepiped," and incompatible financial management systems, SABRS was envisioned to be a comprehensive, user-controlled financial management tool, combining ad-hoc-unit-level report capabilities with real-time transaction processing.

The timetable for SABRS implementation was no less ambitious than its scope. The Automated Data System Development Plan (ADSDP), dated 31 March 1980, called for full system implementation throughout the Marine Corps by October 1, 1982. [Ref. 1] Unfortunately, full implementation did not occur until October 1, 1992 -- ten years later!

Schedule delays are as common to systems development projects today as they were throughout the 14-year development of SABRS. The software engineering literature is replete with books and articles chronicling the challenges associated with developing computer-based systems. What appears lacking in this literature, however, are inquiries into past projects, either successful or unsuccessful, that allow those individuals intimately familiar with a particular project to reflect openly on its development.

Abdel-Hamid and Madnick argue that failure to learn from past efforts has become an enormous obstacle to improving the systems development process. [Ref. 2] They feel strongly that project managers should view mistakes and setbacks as learning opportunities, rather than sources of embarrassment. With this inherent unwillingness to reveal development mistakes on the part of project managers and others, it is not surprising that previous attempts to derive lessons learned from the SABRS project are nonexistent.

B. OBJECTIVE AND FOCUSING RESEARCH QUESTION

The objective of this thesis is to chronicle the SABRS development process via archival research and personal interviews. A central focus is to determine lessons learned about the management of the systems development process.

Specific areas of interest include, but are not limited to, (1) the level of management support provided, (2) the use or abuse of systems development methodologies, and (3) the level of user involvement in the development process.

C. SCOPE, LIMITATIONS AND ASSUMPTIONS

1. Scope

The perspective is from the SABRS project management level; thus, interviews, archival research and literature reviews concentrate on management issues involving systems development. The presentation of data is limited to the SABRS project, although references will be made to the concurrently developed Marine Corps Standard Supply System (M3S).

2. Limitations

The primary limitations encountered in this project were poorly kept documentation and difficulties in locating appropriate SABRS management personnel. Much of the SABRS documentation was either not cataloged or missing altogether, making it difficult to reconstruct development decisions and chronology. Similarly, given the lengthy development period, it was not easy to locate individuals who were knowledgeable in the management of the SABRS project. Despite these limitations, the author does not feel that their absence significantly impacts the data contained in this presentation.

3. Assumptions

This thesis is intended to be read by anyone curious about the systems development process, especially new project managers unfamiliar

with the difficulties and pitfalls that are likely to occur. Although helpful, prior knowledge and experience in systems development is not required.

D. THESIS ORGANIZATION

Chapter I briefly introduces SABRS and orients the reader toward the goal of this thesis: to derive general lessons learned about the SABRS system development process.

Chapter II gives background information on the major systems development theories that influenced SABRS developers, including the waterfall model and the prototyping paradigm. Also listed are a number of factors considered to be causes of development delays, cost overruns, and unfulfilled requirements.

Chapter III details the qualitative research methodology employed by the author. An explanation of the interview process, including a description of the four interviewees, is provided.

Chapter IV familiarizes the reader with specific details pertaining to the origination of the SABRS concept. Additional background material is provided concerning previous Marine Corps financial management systems, as well as the concurrently developed M3S system.

Chapter V uses the acquired documentation to reconstruct important SABRS development events. The organizational structure supporting the

project is also provided, along with a description of the three "eras" of SABRS development.

Chapter VI is a narrative presentation of data obtained by the author via personal interviews with four prominent members of the SABRS development team.

Chapter VII derives lessons learned about the SABRS development process based on the author's observations and analysis of the data obtained. Eight specific lessons are presented and supported with brief discussions.

Chapter VIII summarizes the thesis and offers some specific recommendations for further study.

II. THE SYSTEMS DEVELOPMENT PROCESS

A. INTRODUCTION

To better understand the issues confronting those involved in the SABRS development effort, it is necessary to describe pertinent systems development methodologies and the problems associated with their use. This chapter begins by outlining the major theories that influenced SABRS development. After presenting three generic phases common to all systems development projects, the chapter closes with a discussion of some major factors that cause project delays and cost overruns.

B. THE CLASSIC "WATERFALL" MODEL

1. Overview

As every student of systems development learns, there is a classical approach to building computer-based systems. Sometimes referred to as the systems development life-cycle (SDLC), it is better known today as the "waterfall" model. The waterfall approach seeks to define specific steps (or phases) through which a development must pass in order to successfully complete a project. These phases include requirements analysis, systems analysis, design, coding, testing, fielding the system and maintenance. In theory, these phases are not rigidly sequential and often require some

overlap. Similarly, the waterfall model allows feedback from later phases, giving an opportunity for developers to correct flaws and oversights prior to implementation. [Refs. 3, 4]

Sprague and McNurlin [Ref. 5] detail an excellent list of characteristics most often associated with this classical paradigm. They include:

- Hand coding in a third generation language (such as COBOL)
- A "structured programming" development methodology
- An automated project management system
- A database management system
- A mix of on-line and batch applications in the same system
- Development of mostly mainframe applications
- Programming by professional programmers only
- Various automated (but not well integrated) software tools
- A well-defined sign-off process for system delivery
- User participation mainly in requirements definition and installation phases.

2. Goals

While a development project attempting to use the waterfall model may not exhibit all characteristics, most can be found. A majority of these characteristics are necessary because they reflect the model's threefold

goals: to introduce discipline, improve reliability and reduce errors, and use resources more efficiently. [Ref. 5]

a. Introduce Discipline

The first goal is to introduce discipline into an unstructured process. During the 50's and 60's, when programming and systems development was in its infancy, virtually all software was custom-made. The programmer designed, coded, implemented, operated, and maintained the system. As hardware technology advanced in the late 60's and early 70's, allowing multiprogramming and multi-user environments, software became the focal point of development. The volume of source code required to run such a system was increasing rapidly. Users could no longer perform the myriad of programming tasks necessary to develop these larger systems; thus, dedicated programmers were hired and given the difficult job of translating user needs to functional systems. Perhaps the final straw emerged in the late 70's with the advent of distributed systems that increased program complexity tremendously. Despite this growth in volume and complexity, programmers were still designing systems in their heads, creating systems that were both poorly documented and impossible to maintain. [Ref. 6]

As project delays, backlogs, and cost overruns became commonplace, proponents of the life-cycle approach argued that the only

way to develop complex systems was to define and formalize the development process. Detailed documentation throughout each phase was also required, adding a further layer of structure to a previously haphazard process.

b. Improve Reliability and Reduce Errors

A further goal of the life-cycle model is to improve product reliability and maintainability. To some degree, the waterfall method acknowledges that errors due to oversight cannot be eliminated completely. Feedback loops hypothetically drawn between each stage allow developers to redo system components as soon as mistakes are discovered. These mistakes and oversights are uncovered through a system of detailed inspections carried out during each development phase. In theory, this should allow errors to be corrected at the earliest possible stage. The importance of correcting an error early cannot be overstated. As Boehm [Ref. 7] noted as far back as 1981, if the cost of correcting an error in the requirements phase is \$1, the cost increases by a factor of 100 if that same error is not caught until the operational phase!

c. Use Resources More Efficiently

The final goal of the waterfall model is to foster more efficient use of financial and personnel resources. The structured stages of development provide a "cookbook" approach that most project managers feel

comfortable using. Deadlines, personnel policies, and cost control systems are established to correspond with each development phase. Ideally, these management initiatives result in a more smoothly administered development, reducing the possibility of delays and cost overruns.

Despite its laudable goals, the classic waterfall model has not proven to be a panacea for improving systems development. Many problems have been encountered by those attempting to apply the model to projects in the "real world".

3. Problems

a. Too Much Documentation

Boehm [Ref. 3] notes that a major shortcoming of the waterfall model is the importance placed on detailed documentation, especially its use as a measure of progress during the early requirements analysis and design stages. What was seen by proponents as a means of controlling an unstructured process has become an end unto itself. Rather than stress the importance of accurately capturing user requirements, developers often allow documentation to drive the process. Regardless of its level of detail, documentation that fails to address user needs is both useless and costly.

b. Requirements Are Not Stated Correctly

Another criticism of the life-cycle approach is that users often cannot thoroughly state all requirements during the early development stages. Pressman [Ref. 6] asserts that the waterfall model has difficulty handling the uncertainty found at the beginning of most projects. The tendency of the user is to say, "I'll know what I want when I see it." The sequential nature of the classic life-cycle cannot accommodate these instances. Furthermore, the language of the user community often differs significantly from that of the developer. Getting the developer to thoroughly understand the detailed needs of the user is a time consuming and often impossible task.

c. Too Many Methodologies

The search for ways to overcome the communication difficulties inherent between users and developers has spawned its own industry. Famous names, such as DeMarco and Yourdon, have become systems development icons through their works detailing how to navigate through specific *phases* of the life-cycle. [Refs. 8, 9] Known as methodologies, these works provide specific step-by-step instructions for completing a particular development phase.

One such methodology is "structured analysis," used during the requirements gathering phase of systems development. Unfortunately, there are many versions of the structured analysis methodology, each with

its own set of symbols and guidelines on how to diagram user requirements. Learning one methodology is not sufficient; the chosen analysis method is often based on customer demands and/or the personal preference of the project manager. When methodologies for follow-on phases (such as the numerous versions of "structured systems design") are included in the mix, it is apparent that there are too many confusing variations. As will be shown later in this thesis, project managers who become intoxicated with the benefits promised by each new methodology risk miring their projects in an endless attempt to define requirements.

C. PROTOTYPING

1. Overview

A user may enter the systems development process with a well-defined set of objectives, but be unable to express the desired input, processing or output requirements. In these instances, prototyping may provide the best approach. Pressman offers this summary:

Prototyping is a process that enables the developer to create a model of the software that must be built. The model can take on one of three forms: (1) a paper prototype or PC-based model that depicts human-machine interaction in a form that enables the user to understand how such interaction will occur, (2) a working prototype that implements some subset of the function required of the desired software, or (3) an existing program that performs part or all of the function desired but has other features that will be improved upon in the new development effort. [Ref. 6, p. 27]

The prototyping paradigm differs significantly from the waterfall model. Although requirements gathering is the first step in both, the next step in prototyping is to produce a "quick design." This design concentrates on visually representing inputs and outputs requested by the user. From the quick design, a prototype is fabricated. The user then evaluates the prototype and initiates a process of refinement and iteration. Ideally, these steps allow the developer to better understand the needs of the user. Within a strict interpretation of the methodology, when the revised prototype is accepted, the developer "throws away" the working model, using what was learned about user requirements to design a more robust system.

Although prototyping appears to remedy the problem of accurately defining user needs, its use has revealed a number of disadvantages.

2. Disadvantages

Pressman [Ref. 6] and others are critical of specific aspects of the prototyping paradigm. First, there is often confusion between the developer and the user/customer over the throw-away issue. When the customer sees the "tuned" prototype, he/she may feel the product will soon be ready for implementation. Unfortunately, the software is of little use because it possesses only superficial functionality, focusing instead on visual representations. Upon learning that the system must be reconstructed, the customer insists that further delays are unacceptable and demands that the

prototype be made into a working system. According to Pressman, software development management usually gives in.

A second disadvantage inherent in the prototyping paradigm involves compromises the developer might make to quickly construct a working model. A programming language or operating system inappropriate to the larger project may be used simply because it is familiar and already owned. Furthermore, an algorithm may be employed that is either inefficient or unusable in a full-scale project in order to demonstrate capability. The danger here is that the developer will design the prototype with properties unique to the chosen algorithm, programming language, or operating system, neglecting all the reasons why they were inappropriate. The unsuitable choice is now an integral part of the system. [Refs. 4, 6]

3. Prototyping and Fourth-Generation Languages

The use of the prototyping paradigm has received greater attention with the continued maturation of fourth-generation languages. Third-generation languages, such as COBOL, C, and ADA, rely on the programmer describing in considerable detail exactly what the program is to accomplish. The theory behind fourth-generation languages is that the user/developer specifies *what* is to be accomplished, and the program determines *how* to carry out that task. Ideally, code will be generated automatically by the fourth-generation language translator. As of today, however, few products offer complete fourth-generation capabilities. The most powerful fourth-

generation languages, such as spreadsheets and database programs, operate in very specific domains. These include FOCUS, Linc, NATURAL and others [Ref. 4, 10].

As fourth-generation languages continue to mature and exhibit more robust code-generating capabilities, they will enhance the developer's ability to rapidly construct usable prototypes. In fact, this should also prevent the developer from having to jettison the working prototype in favor of a more powerful third-generation language. As Emery and others [Ref. 11] advocate, this "adaptive methodology" essentially relies on the "...evolutionary development of a prototype program that eventually becomes the operational system...." [Ref. 11, p. 15]. Thus, fourth-generation languages (also called I-CASE) may allow developers to overcome many of the disadvantages that plague the traditional prototyping paradigm.

D. THREE GENERIC PHASES OF SYSTEMS DEVELOPMENT

Regardless of the model chosen (there are many others not germane to SABRS development), the systems development process contains three generic phases [Ref. 6]. These phases include:

- **Definition phase.** Includes systems analysis, requirements analysis, and software project planning. Attempts to identify what needs to be done.
- **Development phase.** Includes software design, coding and software testing. Attempts to determine how the architecture will be designed and translated into a programming language, as well as *how the system will be tested.*

- **Maintenance phase.** Includes error correction, system adaptation, and system enhancement. Focuses on a planned program for implementing *changes* to the system.

E. FACTORS CAUSING DELAYS, COST OVERRUNS, AND UNFULFILLED REQUIREMENTS

1. General

Despite the use of various development methodologies aimed at improving project management, projects continue to suffer delays and cost overruns. Reasons for these shortcomings range from improper use of the chosen methodology to incompetence on the part of technical and management personnel. Unfortunately, the list of causes is broad and constantly changing, making it difficult to establish rules that apply to all projects. The literature, however, does contain some generally accepted factors that inhibit the systems development process. This section summarizes those factors.

2. Shortcuts

Shortcuts taken during the project often result in extensive rework later. Under pressure to fulfill unrealistic deadlines, developers may skimp on requirements analysis, design, and testing to keep the project on schedule. The end product, however, fails to meet customer expectations, thus requiring extensive reconstruction. As described previously, such

rework in the operational phase can be extraordinarily costly and time consuming. [Ref.4]

3. Analysis Paralysis

In contrast to the shortcut problem is the fear of leaving the requirements analysis phase at all. Gause and Weinberg [Ref. 12, p. 277] mimic an Oscar Wilde remark by stating, "After two, or five, or even ten years, you can dip into the ongoing requirements process and watch them take out a comma in the morning and put it back again in the afternoon." The same authors point out that, while developers must have the courage to end the requirements analysis phase, the *process* of refining requirements continues. Most developers and project managers mired in this "analysis paralysis," however, are convinced that if they simply study the problem a little longer everything will miraculously fall into place. Unfortunately, such dogged determination only results in further delays and cost overruns.

4. Users Are Not Involved

Users, the people for whom the system is being developed, are often overlooked as having any significant impact on systems development. On the contrary, their involvement is essential throughout the process. After all, it is the user who must be satisfied with the product for it to be considered a true success [Ref. 12, p.69]. Delays and cost overruns result when user feedback is not sought because requirements are seldom

translated perfectly into the desired system. As with shortcuts in the development process, extensive rework must be performed to correct deficiencies.

5. Unreasonable Demands

Often, upper management will require precise cost estimates prior to fully funding/approving the development effort. This occurrence is especially relevant to DOD systems, where extensive functional and economic detail is mandated even before systems analysis begins [Ref. 13]. The Government Accounting Office (GAO) regularly criticizes DOD systems development for an "... almost total lack of accuracy in cost estimates" [Ref. 14, p. 7]. Unfortunately, it is unreasonable to expect accurate cost estimates before any meaningful, detailed analysis of the system has begun.

6. System Complexity

Dr. Emery introduces complexity as another factor obstructing efficient systems development. He asserts that an information system is often relied upon to coordinate the activities of the organization it serves. As such, the complexity of the organization is mirrored in the complexity of the information system proposed. As the complexity of the organization increases, demand for the information system to provide greater functionality also increases. At some point, the desired requirements will reach or exceed the organization's current systems development capabilities.

Nevertheless, development forges on; and it is no surprise that delays and cost overruns result. [Ref. 11, pp. 2-3]

7. Inexperienced Technical and Managerial Personnel

The Department of Defense is seriously devoid of experienced personnel in both the technical and managerial aspects of systems development. In fact, this problem permeates all Government agencies. A 1989 House staff study [Ref. 15] stated a number of reasons why. First, salaries for computer specialists range from 23 to 32 percent less than those in private industry. Furthermore, experienced senior management salaries are 65 percent behind the private sector. Second, meaningful career paths are nonexistent in some organizations, particularly within DOD, where the culture favors the warrior over the technical specialist. Without established career and educational opportunities, the persistent turnover of qualified personnel that plagues all federal agencies will continue.

It is the author's opinion that this inexperience among DOD systems development personnel and project managers causes problems from the earliest stages of development. For example, when the feasibility of a new system is being considered, the primary architects are functional area experts, not systems development professionals.

Consider an organization within DOD that is determining the need for a new pay system. The initial development team would consist primarily of financial experts. Therefore, those with limited systems development

backgrounds are formulating the very cost and schedule estimates upon which funding and approval systems are based. This creates enormous difficulties later when development functions (systems analysis, design, programming, etc.) are outsourced; what appeared logical to the functional representatives simply cannot be accomplished by the actual developers. Initial assumptions must be reworked. Unfortunately, the original milestones and cost estimates are still used to monitor progress.

F. SUMMARY

This chapter provided necessary background information relating to the theory and problems associated with systems development. The classic waterfall model and the prototyping model were described and analyzed prior to presenting the three generic systems development phases. The chapter closed by listing some primary causes of late systems delivery and cost overruns. The theories and issues presented were selectively chosen by the author to reflect those areas most pertinent to SABRS development.

III. RESEARCH METHODOLOGY

A. INTRODUCTION

In keeping with the qualitative nature of this thesis, the research method employed was the historical case study, relying on multiple sources of data, including both archival material and personal interviews. This chapter describes the collection of SABRS documentation and how interviewees were both selected and questioned.

B. ARCHIVAL DATA COLLECTION

Initial phone conversations revealed that all SABRS related documentation was located at the Defense Finance and Accounting Service (DFAS), Kansas City, Missouri (formerly the Marine Corps Finance Center, Kansas City). Three days were spent in Kansas City reviewing these data. Documentation consisted of eight bookshelves filled with binders pertaining to SABRS development. Unfortunately, none of these data were cataloged and many of the binders did not contain material corresponding to the cover title. This obviously made the research effort somewhat frustrating and time consuming. Furthermore, no data were found relating to the management of the SABRS program, such as Systems Decision Papers and Mission Needs Statements. These life-cycle management documents are required of all

DOD components to defend various development decisions and justify further project funding [Refs. 13, 16]. These data would have proven invaluable, thereby allowing the researcher to quickly determine reasons for specific development delays.

No one currently working on the maintenance of the SABRS system seemed concerned that the development documentation was unorganized and incomplete. Evidently, none of these materials are required for day-to-day maintenance and operation of SABRS. It can only be assumed that missing documents were either: (1) improperly filed, or, (2) lost in transit from Quantico, Virginia in March of 1993, when the SABRS program office was closed and all documentation was moved to Kansas City.

Despite these research difficulties, a number of useful documents were obtained. The original Concept Statement [Ref. 17], Requirements Statement [Ref. 18], Feasibility Study [Ref. 19], and Functional Description [Ref. 20] provide a detailed account of the original specifications, economic analysis and milestones established at the beginning of the SABRS development process. Further documentation relating to general systems analysis and design helped verify the use of structured methodologies [Refs. 21, 22]. None of the documents studied, however, contained any information as to why planned milestones were not achieved.

C. PERSONAL INTERVIEW DATA COLLECTION

1. Selection

The primary difficulty encountered in selecting appropriate interviewees was simply locating persons familiar with broad SABRS development issues. The 14 years taken to field SABRS resulted in many members of the development team serving only during specific stages of development. Turnover of key management decision makers occurred frequently, primarily due to normal military and government service rotations and promotions.

Fortunately, three former program managers and the primary systems architect were contacted and subsequently interviewed. Each of these individuals possessed broad knowledge of the SABRS development process. They expressed many strong opinions; their comments corresponded on some issues and conflicted on others. In hindsight, the interviewees provided an excellent cross-section of viewpoints that included functional, managerial, and technical perspectives.

2. Background

The first interview was conducted with Mr. George John, GM-15, currently the Deputy Director for Accounting at the Defense Finance and Accounting Service, Kansas City. Mr. John, working in various capacities, has been intimately involved in the SABRS project since 1979. Serving first

as an accounting functional area representative responsible for writing requirements documentation, he later served as an interim program manager, becoming familiar with development methodologies and other management issues. In his current capacity, Mr. John has primary responsibility for the day-to-day operation of the fielded SABRS system. Mr. John spoke candidly about SABRS development, yet appeared to choose his words carefully. Furthermore, his executive officer was present during the entire interview, but did not participate.

The next interviewee was Mr. Ralph Powell, currently an analyst working for Computer Data Systems Incorporated. Mr. Powell is retired from both the Marine Corps *and* the Civil Service, with over 35-years experience in Marine Corps financial management. He joined the SABRS development team part-time in 1981 for the purpose of integrating SABRS accounting policies and procedures. By the mid-1980's, Mr. Powell was a full-time member of the development team, eventually becoming the program manager responsible for operational testing and implementation. Mr. Powell was very confident in his assessment and criticisms of the development process, most likely because of his first-hand experience discovering and correcting errors during implementation.

Lieutenant Colonel (now Colonel) Jack Larson served as the SABRS program manager from 1982 to 1987 and was the third interviewee. Colonel Larson graduated from the Naval Postgraduate School's Computer

Systems Management curriculum in June of 1982. He is often credited with providing the leadership that ultimately revived SABRS development in the mid-1980's. The Colonel is knowledgeable and conversant in all areas of both Marine Corps financial management and the systems development process. The author was previously associated with Colonel Larson, which afforded an extremely relaxed and candid discussion of relevant SABRS development issues.

The final interview was conducted with retired Lieutenant Colonel Alan Craig, now a senior systems developer for Computer Sciences Corporation. LtCol. Craig served as the senior systems architect from 1982 to 1989. His responsibilities consisted of translating system requirements into general and detailed systems design, as well as the coordination of all programming tasks. LtCol. Craig was the senior technical member of the SABRS development team. His interview provided an excellent overview of the technical problems often created by managerial decisions. LtCol. Craig was somewhat reserved during the interview, although he answered each question in extreme detail.

3. Interview Outline

For the author to identify common themes and contradictory opinions, it was necessary to focus each interview around the same set of questions. A most useful outline for this purpose was presented by Dr. Lee Gremillion during a lecture at the Naval Postgraduate School in August 1993

[Ref. 23]. Dr. Gremillion is a Senior Consulting Manager for Price Waterhouse, with many years experience focusing on systems planning and development. A portion of his lecture was entitled, "What Influences Delivery Rate?," referring, of course, to the chronic systems development problems discussed in Chapter II. After researching the topic for many years, Dr. Gremillion believes that the following four categories substantially influence the systems development process:

- Organizational Environment
- Project Team
- Development Environment
- Application Characteristics

This outline is further broken down into specific factors affecting each category. The entire outline is reproduced (with annotations by this author) in Appendix A.

The author used the outline to formulate a sequence of questions focusing on specific SABRS development issues. Appendix B lists the questions derived for all interviews.

It is important to note that use of the outline was not meant to test the validity of Dr. Gremillion's work; rather, it afforded the author a concise yet comprehensive method of inquiry into the SABRS development process.

D. DATA ANALYSIS

Data analysis was performed in two stages. The first stage consisted of scouring the archival material for data pertinent to (1) the genesis of the SABRS program, (2) schedules and planned delivery dates, (3) the use of systems development methodologies, and (4) organizational structure.

The second stage involved compiling interview data. Interview notes were "coded" by searching for common themes and contradictory viewpoints. These results were then combined to derive a number of lessons learned about the SABRS system development process.

IV. GENESIS OF SABRS

A. PURPOSE

To fully appreciate the complexity surrounding SABRS system development, the reader must be exposed to the system requirements considered crucial to the consolidation of Marine Corps financial management systems. Therefore, this chapter describes the formulation of the SABRS concept, as well as an overview of the system and its original objectives.

B. BACKGROUND

1. Problems With Existing Systems

The Marine Corps, like so many large organizations in the late 1960's and early 1970's, developed information systems to meet specific functional area requirements. In these early years of computer-based systems, the mere automation of manual functions improved productivity and efficiency *within that functional area*. If an automated budget system was needed, it was designed and implemented; how the system integrated with other financial management systems was an afterthought. Unfortunately, maintaining these separate "stovepiped" systems required costly management attention and specialized technical expertise. These

systems were modified and upgraded continuously to meet ever-changing legal and fiduciary requirements. Likewise, the inability of these systems to efficiently share data produced redundant and often inconsistent management of financial reports.

At the time of SABRS concept formulation in 1978, the Marine Corps maintained several "stovepiped" financial management systems. The first, referred to as the Priority Management Effort (PRIME) Operations Subsystem, supported all major posts and stations. The PRIME system accounted for all base operation transactions, including all civilian labor and labor distribution as required. The second major system, known as the Marine Air/Ground Financial Accounting and Reporting System (MAGFARS), supported all Fleet Marine Force units. This system was designed on a non-accrual accounting basis and, since there are no civilians in operational Marine Corps units, did not account for civilian payroll and labor distribution. Additionally, because all Fleet Marine Force units are tenants on Marine Corps Bases and Stations, MAGFARS was not designed to perform or account for base support functions. [Ref. 18, pp. 4-5]

Along with these two distinct accounting systems, the Marine Corps maintained a Class I Budget System. Class I systems are developed, programmed, coded, and debugged under the direction of Headquarters Marine Corps. These programs cannot be modified without specific authority from the Commandant of the Marine Corps. [Ref. 24, p. 4] The

Class I Budget System, however, was often supplemented by locally developed systems to support specific budgetary requirements. Likewise, many other locally developed systems were produced to support financial reporting and management requirements. With each Marine Corps command developing its own internal financial management reporting system, the resulting training and maintenance requirements were deemed unacceptable. [Ref. 18, p. 5]

The need for local commands to develop and maintain systems specifically tailored to support financial requirements resulted in the Marine Corps formally identifying deficiencies in its assortment of financial management systems. Deficiencies highlighted [Ref. 18, pp. 11-12] include:

- Under normal conditions, a daily transaction took 10 days to process, become reconciled, and then be recorded in the official accounting records. This excessive period of time required that memorandum records be maintained to insure proper control of funds.
- The MAGFARS and PRIME automated update process required an average of 8 to 10 hours of processing time to complete a daily cycle.
- If a manager wanted to see a report in a different format than what was originally programmed, he had to make a special request. It usually took several days before the information became available in the format desired. This deficiency forced the development of a considerable number of site-unique programs that extracted the requisite information and then presented the data in the desired format.

- The systems did not provide timely data at the level required for the field functional managers to effectively manage or make sound decisions about their funds. For example, a maintenance manager could not get the number of labor hours charged to his job in time to adjust his workforce to stay within the authorized dollar limit.
- System logic precluded concurrent processing of multiple activity accounts, thereby forcing the processing of each account in a separate job cycle sequence.
- Due to extensive system modifications to accommodate new and/or changing user requirements, the resources needed to maintain multiple systems reached unacceptably high levels. The time required to implement modifications to an existing system forced financial managers to maintain manual records.
- The systems produced voluminous hard copy output which was (1) costly and (2) difficult to utilize and manage. For example, requests for the status of a single general ledger account required the production of the entire ledger.
- The systems did not provide for the capture of asset depreciation data, property accounting, production and performance measurement, or contract accrual.

2. Formulation of the SABRS Concept

In August of 1978, the Marine Corps Chief of Staff approved the *Concept Statement for a Single Financial Management System* [Ref. 17]. Its purpose was to authorize the commitment of resources to study the feasibility of developing a single financial management system that would correct the deficiencies listed above. The Concept Statement marks the first official document addressing the need for a newly developed standard accounting, budgeting and reporting system, later to be known as SABRS.

Before development of a single system could begin, however, both a Requirements Statement and a Feasibility Study had to be produced. The purpose of the Requirements Statement is to provide a "... definitive written statement of user requirements," as well as a basis for a Feasibility Study of alternative approaches to satisfy those requirements [Ref. 18, p. 1]. The Feasibility Study [Ref. 19, pp. 1-2] identified the following broadly defined approaches to satisfying user requirements:

- Develop a Standard, Accounting, Budgeting and Reporting System.
- Expand existing Operations Subsystem (PRIME).
- Expand existing MAGFARS System.
- Expand existing Allotment Accounting System.
- Retain existing system status quo.
- Utilize existing Financial Systems of other DOD agencies.
- Devise a manual system.

The remainder of the Feasibility Study details reasons why the first alternative, SABRS, was the selected approach and why the other alternatives were not suitable to meeting user requirements.

C. SABRS OBJECTIVES

In addition to recommending that the old systems be replaced, the Feasibility Study established the following definitive SABRS objectives:

- Provide the commander, and the subordinate managers, inquiry capability with a maximum of 15 seconds response time.
- Insure that the status of funds will be current as of the last transaction processed at the local level.
- Insure that all financial data, other than fund status, will be no more than 24-hours old.
- Provide managers with ad hoc reports.
- Reduce training requirements by 20 percent.
- Reduce input errors by at least 50 percent and correctional processing time by 80 percent.
- Reduce memorandum records by 80 percent.
- Reduce implementation time of directed changes to 30 days.
- Reduce hard copy computer input/output by 70 percent.
- Meet all directed systems standards (i.e., GAO, DOD, HQMC, Privacy Act, etc.).
- Make the system capable of direct input/output with other related systems such as the concurrently developed Marine Corps Standard Supply System (M3S).

Despite such emphasis on measurable objectives (i.e., provide a 15-second inquiry response time), the Feasibility Study failed to discuss how these figures and baselines were determined.

Similarly, these objectives relied almost exclusively on the concurrent development of both the Marine Corps Data Network (MCDN) and the Marine Corps Standard Supply System (M3S) [Ref. 25, p. 1], yet the details of how this was to be accomplished were not included in the analysis.

MCDN was a program scheduled for implementation in 1982 that was to provide the upgraded base telephone lines, connecting trunk lines, front-end processors, and other equipment and software necessary to support SABRS' telecommunications needs. [Ref. 22, pp. 7-8]

The Marine Corps Standard Supply System (M3S) program was a corresponding attempt by the supply community to integrate their myriad old systems into a single system. Because every supply transaction normally involves a corresponding fiscal transaction, it was decided that both SABRS and M3S should be designed around a common database and database management system [Ref. 25, p. 2]. It is somewhat surprising that the important details of this integration were not included in the process that was intended to evaluate alternative courses of action.

D. CONCEPT OVERVIEW

The SABRS distributed network concept was based on the expected telecommunications capability of the Marine Corps Data Network. Six Regional Automated Service Centers (RASC), located at major installations throughout the Marine Corps, were to provide necessary mainframe computer processing power. Computer terminals were to be located throughout each command, utilizing 4800 bit-per-second modems communicating with the mainframe over the Marine Corps Data Network. [Ref. 19, pp. 9-31]

A common database was to be located within each mainframe computer. Data elements were to be shared with the Marine Corps Standard Supply System; thus, meetings with M3S personnel were planned throughout the development process [Ref. 19, p. 39]. The common database concept was essential in order for SABRS to allow the one-time capture of supply transactions. Under the old systems, supply clerks entered requisitions into the supply system and then forwarded a paper copy of that transaction to the fiscal office. A fiscal clerk then entered the transaction into the accounting system. Errors were common and reconciliation of those errors was extremely time consuming. Under SABRS, such transactions would be entered only once, and the resulting data were then shared between the systems, lessening both the time required for processing and the number of errors.

In short, SABRS was envisioned to be a distributed network of mainframe computers, maintaining a common database that would allow multiple users to simultaneously input (via modem) transactions directly into the system. Furthermore, the financial manager would have the capability of accessing his or her current status of funds almost immediately, and in the format desired.

E. SUMMARY

The purpose of this chapter was to describe the circumstances surrounding the Marine Corps' decision to initiate SABRS development. Also presented were SABRS objectives and a brief concept overview, allowing the reader to more fully appreciate the system complexity and expectations established by members of the Concept Exploration and Feasibility Study teams. The following chapter on SABRS chronology includes a presentation of the organizational structure responsible for instituting these goals and requirements.

V. SABRS DEVELOPMENT CHRONOLOGY

A. INTRODUCTION

Determining an accurate timeline of significant SABRS development events was a difficult chore. As mentioned previously, specific development decision papers were not located. However, a thorough examination of development documents, in conjunction with interview remarks, revealed a reasonable break-out of important events. This chapter chronicles those events by first presenting the original milestones established in the documentation. With these objectives firmly catalogued, the project is then divided into the actual "eras" of development derived from interview results. Throughout the chapter, the organizational structure supporting SABRS development is described, where appropriate, to accent the role this structure played in formulating SABRS milestones and managing its development.

B. SOURCE DOCUMENT MILESTONES AND ORGANIZATION

1. Concept Statement Milestones

The Concept Statement approved in 1978 established the following four key development milestones: (1) Automated Data Systems Development Plan (ADSDP) approval by 15 March 1979, (2) analysis and

design approval by 30 August 1979, (3) detailed systems design approval by 30 September 1979, and (4) full system implementation by 1 October 1980.

[Ref. 17, pp. 3-4]

To support this bold schedule, the Concept Statement envisioned all design and programming to be accomplished with "... in-house assets" [Ref. 17, p. 5]. These assets were to consist of full-time personnel from both the Fiscal Accounting Division and the Command, Control, Communications and Computers (C4) Division at Headquarters Marine Corps. Other functional assets, such as budget analysts and logisticians, were to be assigned on a part-time basis. No specifics were stated concerning desired personnel qualifications or the number of people required to complete each milestone.

2. Requirements Statement

Although not referred to in the Concept Statement, the Requirements Statement was the next chronologically published document, dated 30 November 1979. The Requirements Statement Work Group that developed this document consisted of the Chairman, eleven representatives from the Fiscal Division, and four representatives from the C4 Division. Their role was to determine, validate, and publish user requirements based on input received via the formal staffing of proposed requirements to each field activity. The Requirements Statement was intended to provide the Feasibility Study Team a basis from which to evaluate systems development

alternatives. Furthermore, the document states that the Feasibility Study Team began work on 1 January 1979. This creates a certain degree of confusion because that date was eleven months prior to publication of the Requirements Statement, upon which the Feasibility Study presumably depended. Furthermore, there is no *formal* indication that these two documents were intended to be developed concurrently. Note also that the due dates established in the Concept Statement have not been met. Both analysis and design and detailed systems design should have been completed by November of 1979. [Ref. 18, pp. 1-17]

3. Feasibility Study

Following on the heels of the Requirements Statement was publication of the aforementioned Feasibility Study, dated 27 December 1979. The Feasibility Study Team responsible for this document consisted of the same personnel involved in developing the Requirements Statement. In addition to evaluating development alternatives, the Feasibility Study Team established the SABRS Automated Data Systems Development Plan Work Group. This new Work Group would consist of 22 full-time and 11 part-time members, mostly from the Fiscal Division. The Feasibility Study document itself acknowledged for the first time that contractor support would most likely be necessary to augment in-house personnel for both the systems analysis and programming portions of development. [Ref. 19, pp. 1-38]

Note once again that no Concept Statement milestones have been achieved. Furthermore, neither this document nor the Requirements Statement provided any explanation for the delay in accomplishing planned tasks.

4. Top Management Roles and Responsibilities

The Feasibility Study is the first document that identifies the top managers responsible for overseeing SABRS development. Unfortunately, their titles are introduced, but not defined. To locate a description of these responsible positions one must forward to the General Design Document dated September 1986! [Ref. 21]

The highest level of management alluded to in the Feasibility Study was the SABRS Steering Committee. This committee was composed of the Fiscal Director of the Marine Corps, the Deputy Chief of Staff for Installation and Logistics (I&L), and the Director of C4 Systems Division. Established to ensure the proper development of SABRS, the committee's responsibilities included: (1) reviewing status and progress, (2) approving courses of action, (3) resolving conflicts, and (4) providing guidance and direction to the Project Management Office (discussed later) [Ref. 21, p. 16].

Concurrent with his role on the Steering Committee, the Fiscal Director of the Marine Corps served as the Functional Manager, establishing appropriate SABRS requirements and objectives. His responsibilities

included the overall management of SABRS under the cognizance of the Steering Committee. [Ref. 21, p. 16]

The System Sponsor was the final top management position. The Accounting Office within the Fiscal Division held this position throughout SABRS development. The System Sponsor's role was to further establish requirements and objectives, while managing the SABRS project with appropriate guidance from the Fiscal Director of the Marine Corps. [Ref. 21, p. 16]

The role these top management positions played in the development of SABRS, especially that of the Steering Committee and Fiscal Director, will be discussed in the following chapter.

5. Automated Data System Development Plan (ADSDP)

According to its purpose statement, the ADSDP was to "... provide decision makers with a basis for deciding whether to approve for development and implementation a standard financial management system...." [Ref. 1, p. 1]. This document repeats much of the information presented in the Requirements Statement and Feasibility Study documents. Also introduced was a plan to break development into four major phases, closely resembling the three generic phases of systems development described in Chapter II. These phases were: (1) Analysis and Design, (2) Development, (3) Implementation, and (4) Evaluation and Maintenance.

The ADSDP then presented revised development milestones, based on the four phases listed above. Whereas the Concept Statement had planned for complete implementation in less than two years, the ADSDP now expected fielding to be completed by October 1982. [Ref. 1, pp. 9-12]

The ADSDP itself was released on 31 March 1980, a full year behind the original Concept Statement schedule. More importantly, however, before "official" development of SABRS could begin, the ADSDP had to be approved by the Steering Committee. Such approval did not occur until 19 May 1981, over one year later. No reasons were given for this delay.

6. Analysis and Design Action Plan

Because so many documents were missing, chronicling SABRS development after March 1980 becomes even more challenging. For example, the author obtained the Analysis and Design Action Plan (Revised), dated 11 September 1981. A later document, however, referred to the original Analysis and Design Action Plan, dated 05 November 1980 [Ref. 21, p. B-2]. The author was unable to locate this document. Revised documentation would have proven more useful had the incorporated changes been annotated.

Fortunately, the revised Analysis and Design Action Plan did provide the first reference to an official SABRS Development Team. Apparently, sometime between approval of the ADSDP and this revision, a

project manager was assigned, as well as a whole host of functional area representatives. The project manager was the same individual assigned as Chairman of the Requirements Statement Work Group, Feasibility Study Team, and ADSDP Work Group. Also introduced for the first time was a list of 19 contractor billets, including five systems analysts, 12 programmers, one documentation specialist, and one data entry clerk [Ref. 22, p. 31]. Once again, no evidence was obtained describing how members were selected, their qualifications, or how the number of personnel required was determined.

Also included in the 11 September 1981 document was a specific reference to the use of data flow diagrams and structure charts. According to the Plan, these structured analysis and design techniques were to be required throughout SABRS development. No guidance was issued explaining how these techniques were to be used, although reference was made to the Yourdon and Constantine book entitled *Structured Design*, Yourdon, Inc., 1975 [Ref. 22, p. 15]. This was the first formal reference indicating the required use of a particular systems development methodology.

C. THREE ERAS OF SABRS DEVELOPMENT

The above chronology and its supporting documentation only introduces the first three years of SABRS development. As mentioned

previously, later publications served only to confuse the researcher by referencing superseded documents. However, a clearer picture of how SABRS development evolved was acquired through the interview process. All interviewees agreed that SABRS development transpired over the course of three distinct time periods.

1. The "Floundering" Era: 1978-1983

Mr. George John, throughout his interview, referred to the early stages of SABRS development as the "floundering" era. The other respondents concurred, and characterized this era as suffering from (1) a disinterested Project Manager, (2) lack of methodology training and enforcement, (3) analysis paralysis, and (4) improper user/functional area involvement. The following chapter presents the interviewee's comments regarding these issues.

2. The Era of Redirection and Progress: 1983-1987

After five years of wasted effort, the Steering Committee, in conjunction with the Functional Manager and System Sponsor, decided to completely restructure the development effort. A Project Management Office was established, and the entire development team was moved from its previous Fiscal Division office, located in Washington, D.C., to its new site in Quantico, Virginia. Perhaps more importantly, the Program Management Office, while answering to the same top management

structure, was now fully supported by the Marine Corps' Central Design and Programming Activity (CDPA). The Marine Corps' systems development expertise resided within this activity. Despite this important qualification, the CDPA was only partially involved during the "floundering" era, assigning a few programmers and analysts to the project. [Refs. 26, 29]

Concurrent with this major organizational move, the project manager was replaced. The new project manager, Col. Larson, is credited by the other interviewees with reviving SABRS development. As will be revealed in the next chapter, his strong leadership, bold enforcement of a standard development methodology, and willingness to incorporate prototyping produced this turn-around.

3. The Testing and Implementation Era: 1988-1992

The final era involved the ultimate implementation of the SABRS system. The four years required for testing and implementation suggest that many difficulties arose during the fielding of SABRS. For purposes of this study, however, the author chose not to concentrate on this portion of development. Although problems encountered during this era may reveal useful insight into earlier development, the scope of such a study would exceed the author's original goals and objectives. The reader need only understand the chronology of this area relative to the other two.

D. SUMMARY

This chapter provided data concerning the chronology of SABRS development. Early documentation presents the researcher with information regarding planned milestones, organizational structure, and other SABRS development goals. These documents, however, do not disclose the difficulties encountered during the development process. The researcher can only infer that problems occurred in light of the obvious delays that transpired throughout the process. Fortunately, the data acquired from personal interviews does provide the information necessary to more completely analyze SABRS development. The following chapter presents these interview data using the outline introduced in Chapter III.

VI. SABRS INTERVIEW DATA

A. PURPOSE

Consistent with the interview questions posed to each respondent, this chapter provides interview results using the outline described in Chapter III.

B. REFERENCES

All comments and opinions contained in this chapter were obtained during the author's interviews [Refs. 26-29].

C. ORGANIZATIONAL ENVIRONMENT

1. Top Management Support

a. Steering Committee

Top management personnel responsible for project oversight and funding approval strongly supported the SABRS development effort. One reason for this support, however, reveals an interesting caveat. Although not mentioned in SABRS documentation, the interviewees stated that the SABRS Steering Committee was in reality a joint SABRS and M3S Steering Committee. Because M3S had been in development slightly longer than SABRS and held more command interest (a supply system is considered an "operational" necessity to battlefield generals, whereas a

financial management system is considered a necessary evil), M3S was considered the priority system. The support SABRS did receive, therefore, resulted from (1) its planned integration with M3S, and (2) the fact that the SABRS Functional Manager was also a member of the Steering Committee.

LtCol. Craig further noted that both the SABRS development team and the M3S team briefed the Steering Committee three times per year. During these briefings, the M3S presentation required so much time that the SABRS briefing was routinely cut short. In fact, so focused was the Steering Committee on M3S development that once M3S development was cancelled circa 1988, the Steering Committee no longer convened. The SABRS project might have met the same fate had it not been for the personal involvement of the Functional Manager, Mr. Tom Comstock.

b. SABRS Functional Manager

In his dual role as Functional Manager and Steering Committee member, Mr. Comstock was able to stress to other Committee members the urgent need for a standard financial management system. All interviewees were of the opinion that having such a strong proponent at the highest level, who understood the strategic necessity of providing an integrated financial management capability to the field, allowed the project to proceed despite its many schedule delays. Mr. Comstock was such a proponent of SABRS that he took the time during the testing and implementation stages to personally visit each installation site. Both Mr.

Powell and LtCol. Craig felt that it was unusual for a member of the Senior Executive Service to display such an interest in the development of a financial management system. However, despite Mr. Comstock's personal interest and commitment to the SABRS project, he was strictly a financial management expert and, therefore, unable to provide guidance to his subordinates concerning systems development matters.

c. C4 Project Officer

The top management representative tasked with providing systems development guidance and review was the Command, Control, Communications, and Computers (C4) project officer. This officer was assigned by the Steering Committee's C4 representative, and both attended all of the tri-annual SABRS briefings.

LtCol. Craig was very critical of the role assigned this individual. As the principal systems architect during the redirection and progress era, LtCol. Craig did not feel that proper reviews of his team's work were performed by the C4 project officer. He felt strongly that such reviews of systems analysis, general and detailed design, and coding would have greatly benefited this project. LtCol. Craig further stated that ". . . nobody from C4 checked on the design, nor were any external reviews performed."

Rather than criticize the individuals assigned, however, LtCol. Craig criticized the C4 project officer selection criteria. He pointedly noted that the C4 project officer was rotated frequently and always filled by newly

graduated Captains from the Computer Systems Management curriculum at the Naval Postgraduate School. Although he complimented the broad education received, LtCol. Craig was quick to argue that these young officers lacked the practical experience required to oversee the detailed technical review of large systems development programs. LtCol. Craig needed someone qualified to perform these technical reviews with both financial management and systems development experience, not further layers of management oversight. Unfortunately, the officers assigned were never financial management or data systems specialists and, furthermore, did not possess any knowledge of software verification or validation procedures. As a result, external reviews of completed analysis, design and coding was simply never done. In fact, the C4 project officer interacted with the systems development team only during the tri-annual Steering Committee briefings.

d. Program Management

Although the support of top management was viewed as crucial by all interviewees, the person deemed most responsible for the success and/or failure of each SABRS development stage was the program manager. The original SABRS program manager, who was also the chairman of the preliminary study teams outlined in the previous chapter, held this position throughout the "floundering" era. He was, perhaps, the most experienced civilian manager associated with Marine Corps

accounting. Unfortunately, according to Col. Larson, SABRS program management was simply "...not his thing." He possessed neither the patience nor a strong desire to learn the systems development process. Throughout his five-year tenure, the project produced a great deal of documentation, but showed no meaningful progress.

During the transition from the "floundering" era to the era of redirection and progress, the original program manager was reassigned and Col. Larson was given SABRS responsibility. Col. Larson set a new direction for SABRS development through his determined leadership style. The other interviewees characterized him as a dynamic leader who combined functional area expertise with a broad knowledge of the systems development process. He did not, however, rule by fiat. He trusted his technical expert, LtCol. Craig, allowing him to make major decisions involving the systems architecture without repeatedly returning to re-do each supporting functional requirements definition. Col. Larson himself pointed to this specific delegation of authority, noting the importance of allowing a technical leader to emerge who is capable of making day-to-day design decisions. Moreover, he felt that the technical leader must rise from within the organization, rather than from contractor support personnel. This provides the program manager a level of confidence that technical decisions are filtered through someone who fully understands the organization for which the system is being developed.

2. User/Systems Development Team Relationship

The process of getting users involved in the early development was considered crucial by the three program managers interviewed. Not only was it important for purposes of accurately defining requirements, it was also necessary to ensure future success during the transition period from the old, comfortable system to the new, unseasoned system.

When asked how users were incorporated into SABRS development, Mr. John responded by explaining the process used during the "floundering" era. Periodic field visits were made to each financial management activity by members of the SABRS development team. Prior to these site visits, advance copies of proposed requirements were mailed to each activity. Upon receipt of these requirements, field comptrollers and accounting officers were to review the proposed requirements and prepare comments for the visiting development team representatives. According to Mr. John, the site visits primarily involved the representative comptroller(s) and the accounting officer. Therefore, Mr. John believed that only the *information users* of the current financial system took part in the field visits. *Actual system users* -- those who entered data, programmed locally required reports, and operated the current systems -- did not participate in the review.

Mr. Powell witnessed the problems caused by this lack of user involvement throughout the testing and implementation phase. He

repeatedly encountered personnel who maintained little or no vested interest in successfully implementing SABRS. Col. Larson validated this difficulty. While serving as the Comptroller of the 3rd Marine Expeditionary Force, subsequent to his tenure as the SABRS program manager, Col. Larson was charged with overseeing the final implementation of SABRS throughout his command. On one occasion, Col. Larson attempted to verify a list of system errors attributed to the new system. Col. Larson described the individual who submitted this list as a veteran user of the old MAGFARS accounting system, who was constantly complaining about having to learn SABRS. Of all the errors chronicled by this individual, over 90 percent were not connected in any way to the performance of SABRS, yet it was determined that to produce such errors, the individual would have had to enter meaningless data or otherwise sabotage the new system.

Col. Larson used this example to stress the importance of getting as many users as possible involved in the earliest stages of development. Without this involvement, many individuals become fearful of the coming change, unwilling to support a system developed by "those in Washington." Even as early as 1983, Col. Larson experienced resistance from alienated users who had already determined that SABRS was destined for failure.

D. PROJECT TEAM

1. Functional Area Qualifications

Each interviewee characterized the "floundering" era as one that suffered from the absence of qualified functional area personnel. Reasons for this deficiency can be divided into two major causes: restructuring of the financial management officer community during the early 1980's, and (2) the unwillingness of field units to give up their best technical people.

a. Restructuring of the Financial Management Officer Community

Mr. John expressed concern that throughout the early analysis and design of SABRS, the Marine Corps was in the process of losing a great deal of its active duty accounting and budgeting expertise. The majority of these seasoned Marines were either Limited Duty Officers (LDO's) or Warrant Officers. A commission or appointment to one of these ranks required that the individual possess considerable experience as an enlisted member. It also signified that this individual had consistently maintained outstanding performance *within his or her specialty field*. A comprehensive and competitive selection process was used to ensure that only the most qualified individuals were selected to fill the limited number of billets allowed by Congress. During the late 1970's, the Marine Corps financial management community consisted predominately of these "restricted line" specialists.

However, during the early 1980's, the Marine Corps began allowing newly commissioned "unrestricted line" Second Lieutenants to choose financial management as their primary Military Occupational Specialty (MOS). The rationale behind this program, from the Manpower perspective, was that such opportunities for young officers would create a pool of qualified financial managers who could, later in their careers, serve as senior comptrollers and disbursing officers. Unfortunately for such projects such as SABRS, both LDO's and Warrant Officers were forced to leave active duty to make room for this new crop of officers. As one might expect, billets on the SABRS development team were often filled with these less experienced unrestricted line officers. Mr. John felt that SABRS suffered because it was impossible to replace 15 or 20 years of functional area experience with officers who possessed less than five.

b. Unwillingness to Give Up Technical Experts

Despite the restructuring of the financial management community, experienced LDO's and Warrant Officers were not totally purged. They occupied numerous technical billets, especially in the Accounting Offices of major Marine Corps installations. Unfortunately, as Mr. Powell adamantly noted, field units were unwilling to release these valuable individuals to serve on the SABRS development team. Furthermore, the "floundering" era project manager failed to raise this issue with either Mr. Comstock or the Steering Committee. According to Mr.

Powell, such attention by these senior leaders could have forced the transfer of a number of key officers to the SABRS project.

Col. Larson, on the other hand, raised this issue during his tenure. Although a few experienced officers did join the team, the MOS restructuring resulted in a limited number of remaining billets for field LDOs and Warrant Officers. Because SABRS was still in development, the old MAGFARS and PRIME accounting systems remained in use. Those restricted line officers left on active duty were the only officers with the requisite expertise to operate these old systems effectively. So, during the first decade of SABRS development, the Marine Corps had created a situation whereby it could not risk crippling its current accounting process in the hopes of developing an already questionable system.

2. Internal Organization

Except for the Concept Statement's initial assumption that all systems development would be performed in-house, SABRS development required three groups of professionals: military personnel, civil service employees, and contractor representatives. It was determined early on that the technical expertise necessary to completely define, analyze, design, and program SABRS could not be performed with the available personnel. Having witnessed the consequences of mistakes made during the "floundering" era, LtCol. Craig was quite outspoken when asked to evaluate the internal organization of the SABRS development team.

Most disturbing to LtCol. Craig was the exclusion of Central Design and Programming Activity (CDPA) involvement during the "floundering" era. Although required during the original Concept Statement, CDPA sponsorship and support was never sought by the original program manager. In the words of LtCol. Craig, Fiscal Division "went on their own with the development of SABRS, without any CDPA assistance."

The original program manager did seek assistance, however, from a systems development contractor. The major criticism by LtCol. Craig of this approach was not the use of the contractor, but rather the project's total reliance on the contractor to perform systems analysis and design. Instead of augmenting systems development by providing the necessary technical expertise, the contractor took control of the development process. From LtCol. Craig's perspective, the original analysis and design performed from 1981-83 was formulated solely with this outside expertise. The contractor hired programmers and analysts with civilian accounting experience who then applied civilian accounting principles to the development of this unique and complex military accounting and budgeting system. Furthermore, because none of the military or civil service functional representatives possessed systems development experience, they did not recognize the danger of complete contractor dependence. Documentation was the only byproduct of this reliance, most of which proved useless.

The Steering Committee finally acted on this issue in 1983 by reorganizing the development team. Although the Fiscal Accounting Division remained the System Sponsor, the Program Management Office now resided within the CDPA at Quantico, Virginia. It was at this point that LtCol. Craig became involved in SABRS development. The CDPA now maintained a vested interest in SABRS and the project manager had a source of in-house technical expertise upon which to rely.

Mr. John, Mr. Powell, and LtCol. Craig praised the effort Col. Larson placed on teamwork within this new development organization. He also worked hard to integrate the separate cultures that are often exposed when military and civilian personnel work closely together. Col. Larson created a feeling among all development team members that SABRS was "their project." In fact, LtCol. Craig noted that, although four major contractors were used during SABRS' 14-year development life, a number of programmers and analysts remained on the project for the duration, asking to be rehired by whichever company was awarded the contract.

3. External Organization

There were only two noteworthy points brought out by the interviewees concerning the issue of external organizational relationships. The first, told by Col. Larson, highlights the virtual anonymity the SABRS project received throughout the rest of the Marine Corps. When asked if General Officers above those serving on the Steering Committee expressed

interest in the development of SABRS, Col. Larson responded by stating, "Are you kidding? Most General's eyes fog over at the mere mention of accounting or financial management systems."

The second point was made by LtCol. Craig, and illustrates a positive aspect of the external relationship, as well as reinforcing previous comments concerning top management support. Col. Larson and LtCol. Craig both realized the danger posed to the project by the frequent rotation of key military members. Normal military tours of duty range from two to three-years. Extending members beyond the normal tour length is, to this day, considered detrimental to the military member's career. To combat this policy, the program manager petitioned the Steering Committee to formally sign letters authorizing the extension of key military members of the design team, such as LtCol. Craig, beyond the normal tour length. This formal letter, signed by two General Officers and a Senior Executive Service Grade Six, became a permanent record in the service member's personnel file. There was no doubt in either officer's opinion that these letters prevented tour extensions from impacting each service member's career. In fact, because the letters were signed by such senior leaders, the individuals may have actually benefited from the added attention.

E. DEVELOPMENT ENVIRONMENT

The development environment elicited the strongest disagreement among those interviewed, specifically concerning the use of structured methodologies.

1. Methodology Followed

a. *A Structured Methodology Was Not Followed During the "Floundering" Era*

As promulgated in the Analysis and Design Action Plan, systems development was to be performed using Yourdon's structured systems development methodology. Mr. John, who was heavily involved in drafting user requirements during the early stages of development, claimed that no attempt was made during the "floundering" era to enforce the usage of this methodology. Mr. John used as an example the March 1982 General Systems Design Document [Ref. 30], which laid out the overall design of SABRS. He noted that all design requirements were written strictly in prose. No evidence appeared indicating that data flow diagrams or structure charts were formulated consistent with the requirements of the Yourdon methodology.

Mr. John expressed concern that the program manager lacked both the patience to learn and the will to enforce the use of the structured method. LtCol. Craig, on the other hand, was of the opinion that the contractor's control of the analysis and design stages fostered this hands-

off approach by program management. Thus, complete trust was placed in the analysis and design techniques used by the contractor because, after all, "they were the systems development professionals." According to the LtCol., no Marine Corps team members during this era truly understood the importance that Yourdon's methodology placed on well-developed data flow diagrams and design structure charts.

In contrast, Col. Larson was more critical of the analysis paralysis that he claimed characterized the "floundering" era. Because the functional representatives were not trained to use the unique symbols integral to structured analysis and design, requirements were translated by the contractor's analysts and programmers directly from the prose. Col. Larson felt that this inability to communicate forced the development team to repeatedly re-address the same issues. He interjected that program managers must adhere to milestone deadlines, with the understanding that it may be impossible to resolve every issue that arises in a given phase, especially the early phases.

Unfortunately, the documentation produced throughout this period proved useless. As stated by LtCol. Craig, no analyst would have been able to construct a meaningful design from the paper generated during the first five-years.

b. Methodology Enforcement During the Era of Redirection and Progress

Each respondent identified this period as one in which the new program manager, Col. Larson, enforced the use of the Yourdon methodology. Mr. John was most impressed with this new commitment, crediting the endorsement of Yourdon's method as the single most important reason that progress was made during this era. Mr. John was an absolute proponent of the structured methodology, expressing confidence that its use was essential to the development of SABRS. To support this assertion, he focused on the concurrent M3S development effort, criticizing its program management team for not committing to a single methodology. Mr. John claimed that the M3S program manager was continually influenced by every new methodology that promised to be the systems development "silver bullet." As a result, the M3S program suffered from a series of stops and starts, with each new method demanding that requirements be redefined.

Mr. John also credited Col. Larson for his insistence on training all members of the development team in the techniques used to produce data flow diagrams and structure charts. A structured analysis and design workshop was even held at Quantico, providing two-weeks of how-to classes for all members of the development team, including contractor personnel.

Col. Larson himself recalled many occasions in which he and other members of the development team gathered to formulate and refine data flow diagrams and structure charts. According to LtCol. Craig, new team members were not allowed to deviate from the use of the structured methodology, regardless of their previous experience or preference.

c. Prototyping

Despite enforcement of the structured methodology, by 1986 SABRS had yet to emerge from detailed design. According to Col. Larson, the lengthy development period, combined with increasing Congressional concern over costly DOD systems development, resulted in the possibility that SABRS could be cancelled. "Needing a victory," as Col. Larson phrased it, he decided to allow the use of prototyping to quickly develop a budget formulation subsystem. He hoped that production of this working prototype would forestall cancellation of the project.

A talented Marine Sergeant began work on the prototype using the documentation derived over the previous three years. The tool used to program the budget module was FOCUS, an early fourth-generation language and development environment. The Sergeant coded the prototype without assistance and continually modified the program as defects were found. Within one year, the entire Fleet Marine Force was using the SABRS budget package to formulate annual budget submissions. Although the subsystem required further refinement, Col. Larson noted that the

prototyping effort allowed program management to confirm both the vitality of the SABRS concept and the quality of the user interface.

LtCol. Craig felt that the success attributed to the prototyping experiment was somewhat overstated. He viewed prototyping from a purely "throw away" perspective, questioning whether a project as large as SABRS could be constructed entirely in this manner. LtCol. Craig and Mr. John estimated the current size of SABRS to be well over 590,000 lines of code. LtCol. Craig suggested that a regimented waterfall paradigm, relying on structured methodologies, provided the only mechanism by which such a large project could be managed and integrated.

Mr. Powell, in contrast, stated the opposite opinion. He expressed disgust at the number of documents generated at every stage of SABRS development, feeling that structured methodology requirements created a situation whereby program management was more concerned with producing documents than developing the system. He referred to a "fascination" with the use of structured methodologies on the part of many members of the development team. He was a proponent, however, of developing prototype subsystems and felt strongly that SABRS would have benefited from the continuous visual refinement that this method offers.

2. Tools

The three program managers were not aware of any specific use of software engineering tools during development. LtCol. Craig, however,

was quite familiar with the concept of development aids and indicated that the NASTEC CASE tool was used during detailed design. He further revealed that NASTEC was realistically only a documentation generator typical of the fledgling tools marketed during the 1980's and that its capabilities were, therefore, limited.

The topic of programming languages elicited a more positive response from the four interviewees. According to Col. Larson, COBOL was considered the Marine Corps' standard language for administrative information systems. Sometime during analysis and design, however, ADABASE was selected to function as the SABRS database management system. Use of this commercial database package negated the original need to scratch-build the database with COBOL. A powerful fourth-generation query language, known as NATURAL, emerged as a more compatible and efficient development environment. NATURAL proved to be much easier to use and learn than COBOL. A single NATURAL command performed functions that would normally require many lines of COBOL code. Furthermore, NATURAL did not require compilation. Thus, results of program designs and updates could be viewed immediately. Col. Larson stated flatly that the Marine Corps standard requiring the use of COBOL was totally unrealistic in light of the specific requirements and goals of SABRS. He felt that relaxation of this standard was crucial to the ultimate fielding of SABRS, reducing both the original programming complexity and future

maintenance difficulties. Of the 590,000 lines of code maintained in the current SABRS system, Mr. John estimates that over 500,000 lines are coded in NATURAL. The remaining 90,000 lines are written either in COBOL (still required for batch processing) or in FOCUS (the original budget formulation subsystem). Mr. John expects the newest release of NATURAL to incorporate batch processing routines, thus allowing his maintenance programmers to convert the remaining 90,000 lines of code.

F. APPLICATION CHARACTERISTICS

1. Interaction With Other Systems

As described previously, SABRS was developed concurrently with the Marine Corps Standard Supply System (M3S). LtCol. Craig specifically noted that both systems were designed around the ADABASE database management system. Common data elements were negotiated during the monthly M3S/SABRS development team meetings. Interface standards were to be developed by M3S and copied by SABRS. According to LtCol. Craig, until late 1987 SABRS development was performed under the assumption that M3S would be fielded first.

Mr. Powell, unfortunately, experienced the effects of this dependence during system testing and implementation. Upon cancellation of M3S in the late 1980's, portions of the SABRS system were already being field tested at Camp Lejeune, North Carolina. Decisions made years earlier

concerning common supply/accounting identifier codes were useless. Data fields designed for 32-bits in the integrated system had to be reprogrammed to match the old supply system format. Similar interface problems continued to surface throughout testing and implementation. Mr. Powell claimed that the vast majority of these problems can be traced directly to SABRS' reliance on the failed M3S system.

2. Degree of Definition

Despite the forced integration of SABRS and M3S, all respondents expressed confidence that the project was well-defined at inception. They stated that the goal of establishing a standard financial management system was understood and remained the driving force throughout development.

Mr. John, however, did reveal a specific example of "scope creep" that he felt slowed development. Plant property had traditionally been accounted for separately from financial accounting in the Marine Corps. According to Mr. John, someone determined during development that it would be "nice to have" plant property incorporated into SABRS. Mr. John did not feel that this addition was necessary, adding that the integration difficulties encountered by the developers far exceeded any potential benefit.

3. Technological Complexity

There was a consensus among those interviewed that the technological complexity of SABRS evolved during the lengthy development

process. Obviously, the explosion of desktop computers had its impact. But, the interviewees talked more about system expectations than hardware considerations. They noted that at the beginning of SABRS development, providing a "state-of-the-art" system was a driving concern. Distributed networks communicating with 4800 bit-per-second modems was very much on the leading edge of technology in the late 1970's. By the mid-1980's, however, those involved in SABRS development were less concerned with taking advantage of the wave of technological advancements, focusing instead on simply "getting it up and running," as Mr. Powell phrased it. So, as the project labored on, "good enough" was established as the technological benchmark.

G. POSTSCRIPT

Although the interviewees were not asked to comment on the ultimate success of SABRS' lengthy development, the author believes that, despite the project's many delays and difficulties, the fielding of SABRS has been a qualified success. If success for the SABRS project is defined as meeting its original 1978 goal of integrating multiple Marine Corps financial management systems into a single, user-controlled and highly integrated system, then the current version of SABRS has indeed achieved those objectives. The author spoke informally with a number of system users and maintenance programmers during the Kansas City visit, all of whom

expressed confidence that the system works remarkably well given its notorious history. Furthermore, the Kansas City staff believes strongly that the Department of Defense should have selected SABRS as a model for its planned DOD-wide financial management system. This expression of support from those who must maintain the system was somewhat surprising in light of the difficulties encountered during development. However, the fact that the system functions as intended convinced the author that SABRS must be considered a success.

H. SUMMARY

This chapter has focused on the ideas and opinions of four key members of the SABRS development team. Their comments regarding SABRS organization and the project team, combined with their insight into the development environment and application characteristics, provided an insider's view into SABRS development. The following chapter presents a series of lessons learned, derived from SABRS development, that will incorporate the author's personal observations and analysis.

VII. OBSERVATIONS AND LESSONS LEARNED

A. INTRODUCTION

Data obtained through the interview process communicates a number of themes common to SABRS development. Through the author's personal observations and analysis, these common themes can be translated into lessons learned. In some cases, even contradictory opinions convey important lessons, especially when confirmed through archival data. This chapter scrutinizes the SABRS development process by presenting key lessons learned.

B. LESSONS LEARNED

1. Top Management Support is Crucial to the Systems Development Process

Despite the Steering Committee's initial focus on the concurrently developed M3S system, senior management's involvement in SABRS development proved crucial to its ultimate success. Without this top management support, especially the support provided by Mr. Comstock, SABRS certainly would have terminated along with M3S.

Mr. Comstock was perhaps the only Steering Committee member who understood the strategic necessity of implementing an integrated financial management system. Rather than view SABRS as a convenient

add-on to M3S, as it appears other Committee members did, Mr. Comstock envisioned SABRS as the cornerstone of future Marine Corps financial management. Furthermore, he articulated that vision through personal involvement. His frequent visits to implementation sites illustrates his keen interest in the development process. These actions essentially said to each member of the financial management community, and to other Marine Corps leaders, "this project is important to the Marine Corps and it's important to me."

Such strong support from the senior Marine Corps financial manager undoubtedly fostered project momentum, allowing SABRS to overcome mistakes and development team inexperience. It also permitted the development team a certain amount of "breathing room" in which organizational learning could take place. This subtle point perhaps reveals why top management support is so critical to bold projects such as SABRS. Without someone to "champion" the cause and run interference for the development team, each mistake and subsequent delay becomes the focus of criticism. Rather than learning from these mistakes, the development team is forced to defend the decision-making process that led to them. Fortunately for those involved in the SABRS project, the persistent determination of Mr. Comstock prevented this distraction. His involvement played a crucial role in allowing the systems development team to learn from each difficulty encountered.

2. The Program Manager's Leadership Abilities Matter More Than His Technical Expertise

Advancements made during the era of redirection and progress can be attributed primarily to the leadership displayed by the program manager, Col. Larson. Unfortunately, a lack of leadership preceded this era and was responsible for almost five-years of wasted effort.

The original program manager failed to exercise leadership in a number of instances. First, he ignored the importance of adding technical experts to the project team early in the development process. Second, he failed to request support from the Central Design and Programming Activity (CDPA), even though such involvement was required by the original Concept Statement. Third, he permitted only superficial user involvement through periodic and inadequate site visits. Finally, the original program manager displayed little interest in enforcing the use of the selected structured methodology by permitting the contractor to control analysis and design.

Col. Larson, in contrast, displayed a thorough grasp of the need for the program manager to positively affect the process, rather than passively administer it. His leadership forged an atmosphere of cooperation among the military, civilian, and contractor employees. No longer was one group controlling development; instead, the specific expertise resident within each group was focused toward unified goals. The Colonel's

insistence on the use of the structured methodology also provides evidence of his effective leadership during his tenure. A standard methodology forced all development team members to communicate using a common format.

Neither of these individuals possessed any prior systems development expertise. According to Cash and Fox [Ref. 31], a successful leader in program management must acquire this expertise prior to taking such a position. They state that a project leader must be able ". . . to seek innovative solutions and anticipate problems before they reach the critical stage." They conclude by insisting that these traits ". . . are difficult to acquire without solid experience working with technology."

This last assertion is not supported in the SABRS case. What differentiated the two project managers was their willingness to learn and the self-confidence to apply that knowledge, not their level of technical experience in systems development. Granted, Col. Larson had been exposed to systems development theory while earning his Master's degree, but he was just as inexperienced as the original project manager. Col. Larson's strength was that he recognized this inexperience and used his leadership abilities to motivate those around him who did possess the technical knowledge. The original program manager, though perhaps acknowledging his own technical inexperience by allowing the contractor to control development, failed by not providing the direction and support necessary to use the technical experts effectively.

3. A Technical Leader Must Emerge From Within the Funding Organization

As presented in the previous chapter, analysis and design responsibilities were placed almost entirely on the shoulders of contractor personnel during the "floundering" era. Without CDPA support, program management was forced to entrust technical decision making to the contractor as well. This unfortunate situation resulted in a useless design that was formulated entirely with outside expertise. There was no one qualified to represent Marine Corps interests who also understood the technical ramifications of the contractor's analysis and design choices.

The changes that ushered in the era of redirection and progress, however, thoroughly addressed this problem. Not only was the project now fully supported by the CDPA, but an experienced systems development professional was assigned to the development team. LtCol. Craig's emergence as the project's technical leader allowed Col. Larson to concentrate on the administrative and "big picture" details associated with program management. The importance of the technical leader's emergence cannot be overstated, especially in light of the previous lesson learned. If it is accepted that a project manager's major role is to exhibit leadership and that he need only possess a limited technical knowledge, then the emergence of a technical leader becomes crucial to the day-to-day decision making process.

This technical leader should also come from within the organization funding the project, especially if it is a military organization. The unique nature of military systems development cannot be quickly learned by outsourced developers. As displayed by the original SABRS contractor, the expectation proved faulty that a Marine Corps accounting system could be designed by analysts and programmers possessing experience only in civilian accounting systems. It demanded a professional from within the Marine Corps, who understood the culture of the organization, to oversee the translation of requirements into a meaningful design.

4. Structured Methodologies Only Help Organize the Process

Despite Col. Larson's enforcement of Yourdon's structured methodology, its use did not provide the breakthrough required to move from the paperwork design to a working system. That impetus was provided by the prototyping effort and the use of the NATURAL fourth-generation language (discussed in later lessons learned).

Arguably, Yourdon's structured methodology only helped the SABRS development team by stipulating procedures to more effectively organize the systems development process. The cookbook approach certainly aided SABRS developers throughout the analysis and design phases by providing uniform techniques for building data flow diagrams and structure charts. Moreover, the structured methodology produced a set of

coordinated documents that surpassed anything generated by the haphazard techniques used in earlier iterations.

Difficulties encountered during the testing and implementation era, however, reveal that the application of the structured method did little to prevent design errors. Furthermore, the use of the structured method did not allow developers to test assumptions and design choices prior to formal coding. Fortunately, the use of NATURAL permitted programmers to quickly correct these errors as they were identified. Had such a language not been utilized, each error would have required the complete redesign of previously developed data flows and structure charts. There would be no alternative. The very nature of the structured method, with its systematic progression through each phase, cannot accommodate changes without altering the supporting paperwork. So, while the structured method may have helped in the establishment of consistent development procedures, it offered SABRS developers no direct and efficient pathway to design verification or system implementation.

5. Adaptive Prototyping Can Save a Project

The use of systems development methodologies evolved over the course of SABRS development. This evolution began in the "floundering" era without the use of any methodology. As the project was redirected, strict application of Yourdon's structured method ruled the day. Then, as

pressure to avoid project cancellation mounted, prototyping was called upon to produce a working subsystem.

The focal point of this evolution was the decision to prototype the Budget Formulation Subsystem. This decision was forced upon SABRS developers because of questions surrounding the program's lengthy development. Fortunately for SABRS proponents, Col. Larson possessed the courage to attempt prototyping. What he did not realize, however, was the extent to which the rest of the project relied on the adaptive use of this methodology.

It is important first to note that the original Budget Formulation Subsystem was not constructed according to strict prototyping rules. Instead, the talented Sergeant continually revised his early design, never resorting to "throwing it away" in favor of a more robust version. Adding functionality and correcting coding errors merely became part of his maintenance process. This built-in acceptance of change was made possible because the subsystem was developed in FOCUS, an early version of a fourth-generation language.

Perhaps missed by LtCol. Craig in his reluctance to support prototyping was the realization that use of this adaptive prototyping methodology actually spilled over into the other subsystem efforts. The use of NATURAL fostered, and probably demanded, this approach. Adding functionality and correcting structured design flaws became an integral part

of the development process, just as it was throughout construction of the Budget Formulation Subsystem.

Structured methodology proponents would argue that adding functionality and correcting errors so late in the process indicates a failure on the part of SABRS developers to accurately define requirements and verify design at an earlier stage. What SABRS developers learned, however, was that to produce a well-integrated system, requirements sometimes had to be defined and implemented on the fly. Structured methods cannot accommodate such changes in later phases. Adaptive methods thrive on them.

So, in effect, SABRS owes its ultimate success to the willingness of the program manager to risk prototyping and the subsequent adaptation of prototyping to a project mired in the structured methodology. Unfortunately, the SABRS project had to experience the pressure of cancellation before risking this approach. The challenge, therefore, is to convince program management that such a risk is worth taking early in the development cycle, before pressure becomes a catalyst.

6. Get the Right People Involved

The importance of achieving the correct fit between people and tasks was evident throughout the SABRS development process. In some instances, appropriate individuals were assigned vital roles; in others,

finding the right person for a particular position was simply not a priority. Two examples of this latter case immediately come to mind.

First, and perhaps most frustrating to technical members of the SABRS development team, was the assignment policy regarding the C4 project officer. The skills and experience of the officers appointed never coincided with the responsibilities assigned. As a result, the SABRS development team never received external systems development guidance, nor were their analysis and design choices ever challenged by the intended verification and validation process. LtCol. Craig's comments reveal that external guidance and feedback were not only crucial to the development process, but requested by the developers. Apparently, top management was either unaware or unwilling to alter the established billet criteria in order to obtain personnel with skills more suited to this task.

The second example of the project not matching skills to tasks was also visible throughout the development process. The reluctance of the original program manager and the inability of subsequent program managers to staff the SABRS development team with Limited Duty Officers and Warrant Officers, whose financial management systems experience was invaluable, severely impeded the systems development process. Not only would their expertise have provided a more accurate definition of requirements early in the development cycle, but their influence and support might have eased the resistance to change that flared during system

implementation. Unfortunately, without these individuals, program management was forced to employ either inexperienced young officers or, as was often the case, use civilians with no operational Marine Corps financial management experience.

In contrast to the previous examples, many SABRS development team members fit their tasks quite well. Obviously, Col. Larson proved to fit well in his role as the principal systems architect. Even the talented Sergeant who developed the Budget Formulation Subsystem was tasked with responsibilities that matched his abilities, motivation, and experience. Fortunately, top management was made aware of the importance of extending the tours of those individuals who were best suited to their task, such as LtCol. Craig. Although none of these individuals was a systems development "superstar," their consistent performances and dogged determination helped shape the course of SABRS development. Had the importance of getting the right people involved been an initial priority, useless documentation and requirements deficiencies may not have characterized the first five years of SABRS development.

7. Do Not Become Dependent on an Uncertain Resource

The requirement to concurrently develop and fully integrate SABRS and M3S almost killed the SABRS program. Because M3S was in a perpetual state of analysis paralysis, nothing aside from documentation was ever produced. After nearly a decade of failures, the program was

finally cancelled. Since SABRS was almost totally dependent on the implementation of M3S, it almost suffered the identical fate. As was described earlier, the prototyping effort helped confirm the viability of the SABRS concept and its usefulness as a system separate from M3S.

Despite this new confidence in the viability of the SABRS concept, the project almost buckled under the weight of the integrated design once testing and implementation began. The sudden shift from 32-bit fields to those of the old supply system, along with the need to reprogram each supply interface, stalled testing and implementation. Fortunately for those involved in SABRS development, the use of NATURAL permitted developers to easily (relative to COBOL) reprogram major portions of the supply system interface. As LtCol. Craig commented, up until its actual cancellation, SABRS continued as if M3S would be fielded first. This statement implies that there were no M3S related contingency plans built into the development of SABRS. Documentation reveals, however, that as early as 1981, the Analysis and Design Action Plan [Ref. 22, p. 8] addressed the possibility that SABRS could be ready prior to M3S. Unfortunately, those contingencies failed to account for the one event that did take place: cancellation of M3S.

The events described above highlight the incredible increase in system complexity that accompanies concurrent systems development. It was hard enough for SABRS developers to establish their own development criteria without having to consider the effects of M3S integration. In effect,

the Marine Corps was attempting to construct two different information systems that both mirrored the complexity of the entire organization. The unbridled desire to provide greater functionality, however, forced management to demand that these systems be fully integrated and concurrently developed. But, just as Dr. Emery hypothesized in reference 11 (see Chapter II, section E), the desired functionality exceeded the organization's systems development capabilities. Nonetheless, development pressed on, with both projects experiencing years of frustration and delays.

8. Do Not Be Afraid to Start Over

Arguably, the most important event that took place during SABRS development was the decision to completely redirect the project in 1983. It is most remarkable that this hierarchically driven organization, with its inbred stubbornness and determination, would alter its commitment to the chosen course of action. In this case, the SABRS course of action can be considered all the events that characterized the "floundering" era.

Staw and Ross [Ref. 32] have extensively researched the organizational propensity to commit to failing courses of action, sometimes referred to as the study of escalation situations. They note that the natural tendency of organizations, when forced to reexamine a course of action due to questionable or negative outcomes, is to persist in the original course of action rather than withdraw. In most cases, the commitment actually escalates at an alarming rate. Psychological, social, and structural

determinants all play a role in causing management to persist in this failing course of action.

SABRS development can be considered unique because management did not remain committed to the original, failing course of action. Although their goal of creating an integrated financial management system remained constant, top management did not hesitate to replace the key decision makers responsible for the lack of progress that characterized the first five years of SABRS development. Senior management may be open to some criticism for not recognizing the need for redirection earlier, but the mere fact that such a courageous decision was made warrants praise.

What probably fostered this organizational flexibility and resolve was that SABRS development was not initially identified as being vital to the future of the Marine Corps. Although SABRS' strategic necessity was understood by Mr. Comstock from its inception, he did not use that argument until system cancellation was threatened later in the development process. At the time of redirection in 1983, SABRS had not been "institutionalized," a term used by Staw and Ross in their work on escalation theory. Institutionalized projects rarely undergo reexamination, the authors conclude, because they are so closely identified with the organization. Staw and Ross offer Lockheed's notorious L1011 Tri-Star civilian airliner program as an example. Despite a decade of enormous losses, Lockheed persisted

with its major foray into commercial aviation because the corporation did not want to be known *only* as a defense contractor. SABRS, although obviously on a lesser scale, was fortunate not to be institutionalized in a similar manner. Therefore, without SABRS being integrally associated with the overall purpose of the Marine Corps, senior management felt free to alter the failing course of action in favor of a prudent alternative.

VIII. SUMMARY AND RECOMMENDATIONS FOR FURTHER STUDY

A. SUMMARY

This thesis presented a broad overview of the main events surrounding the fourteen-year development of the Marine Corps' Standard Accounting, Budgeting and Reporting System (SABRS). Its goal was to derive lessons learned about the SABRS systems development process by means of qualitative research techniques that relied exclusively on archival documents and personal interviews.

Following the introductory chapter, the author presented background information relating to the systems development process. The specific theories presented included the two major paradigms that influenced SABRS development: the classic waterfall model, including the structured methodologies designed to navigate developers through each phase, and the prototyping approach. Also detailed were some major factors that are said to cause project delays, cost overruns, and unfulfilled requirements, such as the level of system complexity, the lack of user involvement in the development process, and the inexperience of both technical and managerial personnel.

Chapter III described in considerable detail the methodology used to uncover specific information about SABRS development. Documentation,

although incomplete, was located at the Defense Finance and Accounting Service's Kansas City Office. Further inquiry into the development process required that the author interview four major players associated with the SABRS project, including three former program managers and the primary systems architect.

Chapter IV provided the reader background information into circumstances culminating in the Marine Corps' decision to construct a single, integrated financial management system. The old "system" of financial management consisted of several highly segregated and incompatible systems that required considerable maintenance, while promoting inefficiencies. The goal of SABRS was to address these issues by building a system that would not only integrate across financial management functions, but also provide the user real-time analysis and report generation capabilities.

The next chapter on SABRS chronology examined the early development source documents obtained from the author's visit to Kansas City. An analysis of these documents established a confusing picture of the preliminary phases of SABRS development. Timelines for implementation presented in early documents were subsequently revised without explanation.

Chapter VI closes by characterizing the three "eras" of SABRS development. The first, referred to as the "floundering" era, was marked by

program manager disinterest and seemingly endless analysis, and lasted from 1978 to 1983. The second era, labeled the era of redirection and progress, was characterized by dynamic leadership and increased technical support. The third and final era lasted from 1987 until complete system implementation in 1992. This era was not specifically chronicled, although problems encountered during the testing and implementation period were often tied to decisions made in earlier eras.

Chapter VII presented data obtained from each of the four interviews conducted. Interviewee comments and opinions were interwoven into the text using an outline suggested by Dr. Lee Gremillion during a lecture given at the Naval Postgraduate School. Thus, each respondent's comments on SABRS' organizational environment, project team qualifications, development environment, and application characteristics were summarized based on this framework.

The results obtained from each interview, combined with a thorough examination of the available documentation, enabled the author to generate eight specific lessons learned about the SABRS development process. These eight lessons were presented in Chapter VII and supported by the author's observations and analysis. The key lessons learned are summarized as follows:

1. The support of top management is crucial to developing a successful system.
2. The program manager must be a leader, not a technical expert.
3. A technical leader must rise from within the funding organization to assist the program manager.
4. The only benefit of structured methodologies is that they help organize the development process.
5. The adaptive use of the prototyping paradigm can save an otherwise doomed project.
6. The right people must be fit to the right task.
7. A systems development project should not depend on an uncertain resource/concurrently developed system.
8. An organization committed to a particular course of action must be willing to alter that course when questionable or negative outcomes arise.

B. RECOMMENDATIONS FOR FURTHER STUDY

The author discovered early in the data gathering process of this thesis that it is impossible for one person to cover every issue in such a short period of time. There is still a great deal to be learned from the SABRS project. Therefore, a few personal recommendations may help focus future research for those interested in pursuing this topic.

- **Examine in detail the SABRS testing and implementation era.** As alluded to previously, there were difficulties encountered during this phase that were attributed to earlier design decisions. How were these difficulties corrected? What field activities experienced the most resistance to SABRS implementation? The least?

- **Compare and contrast the maintenance costs of the old MAGFARS/PRIME systems and SABRS.** Cost savings resulting from the elimination of multiple systems is often predicted as part of the new system's economic analysis. Since SABRS has completely replaced the old systems, a direct comparison of these costs should be possible. Similarly, it might be interesting to compare the expected SABRS costs developed in the early 1980's with actual system costs.
- **Study the systems development of the failed Marine Corps Standard Supply System M3S.** Because SABRS was so heavily tied to M3S development, yet SABRS survived while M3S was cancelled, an examination of that failed development might prove useful. Was M3S an "institutionalized" Marine Corps project? What development methodologies were attempted? Was the organizational environment similar to SABRS, or was it significantly different?
- **Thoroughly investigate the prototyping attempt that resulted in the successful development of the SABRS Budget Formulation Subsystem.** The talented Sergeant referred to in this thesis is currently a Civil Service employee maintaining the SABRS system at DFAS, Kansas City. His personal recollections, along with an evaluation of the FOCUS fourth-generation language, should prove enlightening and valuable to future developers.

APPENDIX A

Factors That Influence Systems Delivery Rate

(Excerpted from Ref. 23--annotations are italicized)

I. ORGANIZATIONAL ENVIRONMENT

A. Top Management's Role

1. **Support:** *The degree to which top management supports and encourages the project can directly impact its timetable for implementation. Often, a senior manager becomes the project's "champion," extolling its virtues and strategic necessity to other top managers. Without such strong support, the project risks losing its personnel and funding priority within the organization.*

2. **Involvement:** *In order for top management to be supportive it must also be involved in the development process. Granted, the development team requires a certain degree of decision making power, but top management must be kept abreast of the development's progress. Thus, when problems and delays do arise, top management will be better informed and maintain a vested interest in providing continued support.*

B. User/Information Systems Relationship

On the other end of the spectrum, the user must also possess a vested interest in the development. The IS development team, therefore, should seek user input at every opportunity. Lack of user involvement will probably result in the end product failing to meet user needs, thus requiring extensive rework which is both costly and time-consuming. Similarly, user involvement early in the development process may alleviate the resistance to change encountered during system implementation.

II PROJECT TEAM

A. Individuals Assigned to the Team

1. **Ability:** The talent and competence of all members of the development team will directly affect the project's delivery rate. Obviously, all programmers and analysts will not be "superstars." Given this, the following factors take on greater importance.

2. **Experience:** Prior development experience (or lack thereof) of both management and technical personnel also impacts the systems delivery rate. Experienced developers are better able to anticipate problems and adjust to new ones.

3. **Motivation:** The collective motivation of the development team is a somewhat intangible factor, yet it may overcome inexperience and lack of talent. Each individual's drive and professional pride determine the team's flexibility and resolve in the face of setbacks.

B. How the Team is Organized

1. **Internally:** How the project manager organizes the development team contributes to productivity and harmony. This is especially important in DOD systems development where there is a need to balance Military members, Civil Service employees and contractor personnel. A strong leader is essential to fostering teamwork amongst these diverse groups.

2. **Externally:** Once again, there must exist a continuous dialogue between the user community, the development team and top management. The organizational structure should allow for information to flow smoothly among these three groups. The proper balance between centralized and distributed control and decision making authority is essential in order to maintain communication and flexibility during the development process.

III. DEVELOPMENT ENVIRONMENT

1. Methodology Followed: *The degree to which a systems development methodology is followed may help or hinder the entire process. The development team's willingness to conform to the selected methodology also plays a significant role.*

2. Tools Used: *Today, Computer Aided Software Engineering Tools (CASE) greatly improve the systems development process by automating many time consuming tasks. Specialized CASE software can translate user requirements into specific designs and, in some cases, generate code automatically. Despite this, CASE tools did not perform a significant function during the development of SABRS due to the immaturity of CASE technology in the early-to-mid 1980's; therefore, their use is not chronicled in the report. Fourth-generation languages, on the other hand, are also considered development tools. The use of such a tool can speed the development process, especially when coupled with the prototyping methodology.*

IV. APPLICATION CHARACTERISTICS

1. Interaction With Other Systems: *Complexity of systems development increases dramatically when the system being developed must interact with several other systems. Each interface creates its own set of challenges and limits design flexibility.*

2. Degree of Definition: *The extent to which the users and developers can specifically state the requirements of the system (and stick to them) directly impacts the speed of development. "Gold plating" system requirements increases complexity, thereby stalling the entire process.*

3. Technological Complexity: *The complexity of systems development is affected by the technological desires/requirements of the user. If the user wants the new system to be on the leading edge of technology, he or she is forcing the complexity of the system to increase markedly.*

4. Business Complexity: *The overall complexity of the organization also impacts the systems development process. As mentioned in Chapter II, the complexity of the business is often reflected in the complexity of its core information system.*

APPENDIX B

Focusing Interview Questions

A. ORGANIZATIONAL ENVIRONMENT

1. Explain top management's role in the development of SABRS. Were they supportive? Was there a particular person who sticks out in your mind as being the "champion" of the project?

2. Were any members of the financial community involved in the early analysis and design of SABRS? Were users at the field level asked to participate in the process? If so, were they potential users of the system, users of the data that the system produced, or both?

B. PROJECT TEAM

3. Describe the ability, experience and motivation of the members of the SABRS development team. In general, how did these factors contribute to or hinder the development process?

4. Explain how the development team was organized internally, especially in light of the three communities: military, civil service and contractor support? Were any problems caused by this arrangement?

5. Describe the development team's external relationships with members of the Marine Corps financial community and top management. Was there a smooth flow of communications in each direction?

C. DEVELOPMENT ENVIRONMENT

6. Was a particular systems development methodology used in the development of SABRS? If so, how strictly was it adhered to? Do you feel that use of this methodology hastened or impeded the development process? Explain.

7. Were any systems development tools used during systems analysis and/or design? What programming language was selected? Why?

8. Did documentation requirements, either driven by the selected methodology or government directives, affect the development process? Explain.

D. SABRS APPLICATION CHARACTERISTICS

9. Was SABRS designed to "stand alone" or interact with other information systems? If designed to interact, was such integration planned originally, or did it creep in during analysis and/or design?

10. Was the goal of the SABRS system well-defined initially or did "scope creep"/"gold plating" occur? Explain

11. Was the goal of the system to provide state-of-the-art technological capability or did "good enough" suffice?

LIST OF REFERENCES

1. Marine Corps Standard Accounting, Budgeting and Reporting System (SABRS), *ADS Development Plan*, 29 May 1980.
2. Abdel-Hamid, Tarek K., and Madnick, Stuart E., "The Elusive Silver Lining: How We Fail to Learn from Software Development Failures," *Sloan Management Review*, v. 32, pp. 39-48, Fall 1990.
3. Boehm, Barry W., "A Spiral Model of Software Development and Enhancement," *Computer*, v. 21, pp. 61-72, May 1988.
4. Whitten, Jeffrey L., Bentley, Lonnie D., and Barlow, Victor M., *Systems Analysis & Design Methods*, 2nd ed., Richard D. Irwin, 1989.
5. Sprague Jr., Ralph H., and McNurlin, Barbara C., *Information Systems Management in Practice*, 3rd ed., Prentice-Hall, 1993.
6. Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, 3rd ed., McGraw Hill, 1992.
7. Boehm, Barry W., *Software Engineering Economics*, Prentice-Hall, 1981.
8. DeMarco, Tom, *Structured Analysis and System Specification*, Prentice-Hall, 1978.
9. Yourdon, Edward, *Modern Structured Analysis*, Yourdon Press, 1989.
10. Emery, Dr. James C., *The Strategic Implications of a Productive Software Development Process*, Paper presented at the Workshop on Information Technologies and Systems, Sloan School of Management, Massachusetts Institute of Technology, 14-15 December 1991.
11. Emery, Dr. James C., Gremillion, Dr. Lee L., and Chai, Sui Mei, *The Software Development Process: The Strategic Importance of Getting It Right*, Unpublished working paper, 1991.

12. Gause, Donald C., and Weinberg, Gerald M., *Exploring Requirements: Quality Before Design*, Dorset House Publishing, 1989.
13. Navy Data Automation Management Practices and Procedures, *Systems Decisions*, NAVDAC Publication 24.2, 9 March 1983.
14. U.S. Congress. House. Committee on Government Operations, *DOD Automated Information Systems Experience Runaway Costs and Years of Schedule Delays While Providing Little Capability*, Report submitted by Mr. Conyers, 101st Congress., 1st Session, 10 November 1989, H.R. 101-382.
15. U.S. Congress. House. Committee on Science, Space, and Technology, *Bugs in the Program: Problems in Federal Government Computer Software Development and Regulation*, Staff Study by the Subcommittee on Investigation and Oversight, 101st Congress, 1st Session, September 1989.
16. DOD Directive 8120.1, *Life-Cycle Management (LCM) of Automated Information Systems (AIS's)*, 14 January 1993.
17. Headquarters, United States Marine Corps, *Concept Statement for Single Financial Management System*, 15 August 1978.
18. Headquarters, United States Marine Corps, *Requirements Statement for an Information System to Provide a Standard Financial Management System for Use by All Authorized Operating Budget/Allotment Holders Within the Marine Corps*, 30 November 1979.
19. Headquarters, United States Marine Corps, *Feasibility Study of Approaches to Provide a Standard Financial Management System to be Operated by All Operating Budget and Allotment Holders Within the Marine Corps*, 27 December 1979.
20. Headquarters, United States Marine Corps, *Functional Description (FD) for an ADS to Provide a Standard Financial Management System for Use by All Authorized Operating Budget/Allotment Holders Within the Marine Corps*, 31 March 1980.
21. Marine Corps Standard Accounting, Budgeting and Reporting System (SABRS), *General Design Document*, September 1986.

22. Marine Corps Standard Accounting, Budgeting and Reporting System (SABRS), *Analysis and Design Action Plan (Revised)*, 11 September 1981.
23. Gremillion, Dr. Lee, *Application Development Productivity: Sustaining a Higher Delivery Rate*, Lecture given at the Naval Postgraduate School, Monterey, California, August 1993.
24. Headquarters, United States Marine Corps, *Data Requirements Document for an Information System to Provide a Standard Financial Management System for Use by All Authorized Operating Budget and Allotment Holders Within the Marine Corps*, 25 March 1980.
25. Headquarters, United States Marine Corps, *Economic Analysis of an ADS to Provide a Standard Financial Management System for Use by All Authorized Operating Budget/Allotment Holders Within the Marine Corps*, 27 March 1980.
26. Interview between George John and the author, 18 October 1993.
27. Interview between Ralph A. Powell and the author, 16 December 1993.
28. Interview between Jack D. Larson, Col., and the author, 17 December 1993.
29. Interview between Al Craig, LtCol. (Ret.), and the author, 17 December 1993.
30. Marine Corps Standard Accounting, Budgeting and Reporting System (SABRS), *General Systems Design Document*, 03 March 1982.
31. Cash, Charles, and Fox, Robert, "Elements of Successful Project Management," *Journal of Systems Management*, v. 43, pp. 10-12, September 1992.
32. Staw, Barry M., and Ross, Jerry, "Behavior in Escalation Situations: Antecedents, Prototypes, and Solutions," *The Evolution and Adaptation of Organizations*, Ed. Barry M. Staw and L.L. Cummings, JAI Press, pp. 191-230, 1990.

INITIAL DISTRIBUTION LIST

- | | | |
|----|---|---|
| 1. | Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145 | 2 |
| 2. | Dudley Knox Library
Code 52
Naval Postgraduate School
Monterey, California 93943-5002 | 2 |
| 3. | Commandant of the Marine Corps
Code TE 06
Headquarters, U.S. Marine Corps
Washington, D.C. 20380-0001 | 1 |
| 4. | Professor James C. Emery
Code AS/Ey
Naval Postgraduate School
Monterey, California 93943-5000 | 2 |
| 5. | Associate Professor Nancy Roberts
Code AS/Rc
Naval Postgraduate School
Monterey, California 93943-5000 | 1 |
| 6. | Captain Jeffrey L. Tavares
611 Shamrock
O'Fallon, Illinois 62269 | 2 |